HOME WORK 3

```python
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader
from torchvision import datasets, transforms
import torch.nn.functional as F
import matplotlib.pyplot as plt
import numpy as np
from torch.utils.data import SubsetRandomSampler
from PIL import Image
import random
```

RESNET20

```python
def conv3x3(in_planes, out_planes, stride=1):
    return nn.Conv2d(in_planes, out_planes, kernel_size=3,
stride=stride, padding=1, bias=False)

class BasicBlock(nn.Module):
    def __init__(self, in_planes, planes, stride=1):
        super(BasicBlock, self).__init__()
        self.conv1 = conv3x3(in_planes, planes, stride)
        self.bn1 = nn.BatchNorm2d(planes)
        self.conv2 = conv3x3(planes, planes)
        self.bn2 = nn.BatchNorm2d(planes)

        self.shortcut = nn.Sequential()
        if stride != 1 or in_planes != planes:
            self.shortcut = nn.Sequential(
                nn.Conv2d(in_planes, planes, kernel_size=1,
stride=stride, bias=False),
                nn.BatchNorm2d(planes)
            )

    def forward(self, x):
        out = F.relu(self.bn1(self.conv1(x)))
        out = self.bn2(self.conv2(out))
        out += self.shortcut(x)
        out = F.relu(out)
        return out

class ResNet(nn.Module):
    def __init__(self, block, num_blocks, num_classes=10):
        super(ResNet, self).__init__()
        self.in_planes = 16

        self.conv1 = conv3x3(3, 16)
        self.bn1 = nn.BatchNorm2d(16)
        self.layer1 = self._make_layer(block, 16, num_blocks[0],
stride=1)
        self.layer2 = self._make_layer(block, 32, num_blocks[1],
stride=2)
        self.layer3 = self._make_layer(block, 64, num_blocks[2],
stride=2)
        self.linear = nn.Linear(64, num_classes)

    def _make_layer(self, block, planes, num_blocks, stride):
        strides = [stride] + [1]*(num_blocks-1)
        layers = []
        for stride in strides:
            layers.append(block(self.in_planes, planes, stride))
            self.in_planes = planes
        return nn.Sequential(*layers)

    def forward(self, x):
        out = F.relu(self.bn1(self.conv1(x)))
        out = self.layer1(out)
        out = self.layer2(out)
        out = self.layer3(out)
        out = F.avg_pool2d(out, out.size()[3])
        out = out.view(out.size(0), -1)
        out = self.linear(out)
        return out

def ResNet20():
    return ResNet(BasicBlock, [3, 3, 3])
```

```
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])
```

```
train_data = datasets.CIFAR10(root='./data', train=True,
download=True, transform=transform)
test_data = datasets.CIFAR10(root='./data', train=False,
download=True, transform=transform)
```

```
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to
./data/cifar-10-python.tar.gz

100%|██████████| 170498071/170498071 [00:06<00:00, 25408631.18it/s]

Extracting ./data/cifar-10-python.tar.gz to ./data
Files already downloaded and verified
```

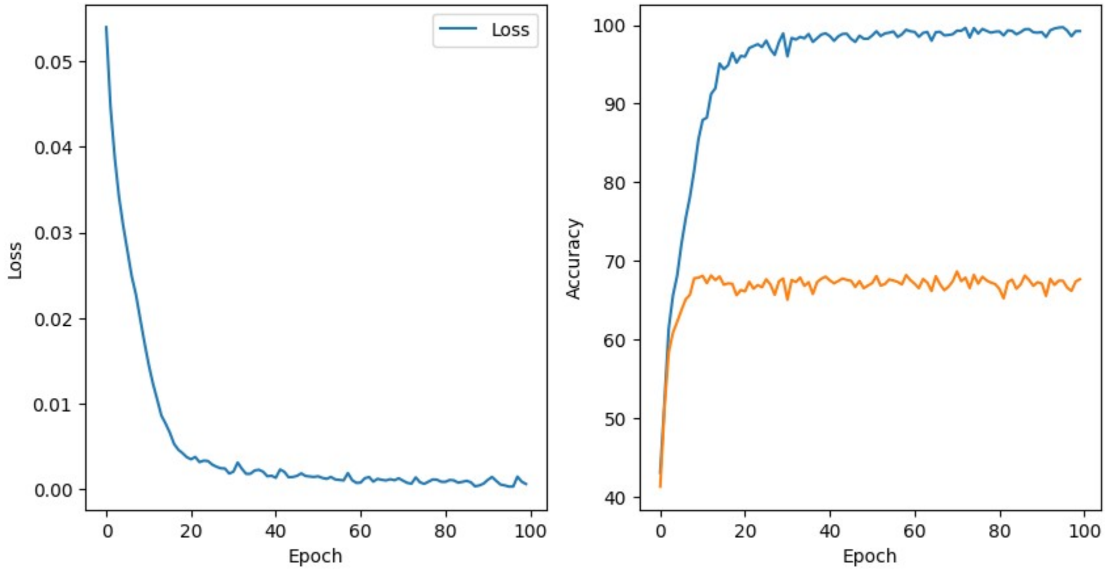
```
subset_indices_train = []

# For each class
for i in range(10):
    class_indices = np.where(np.array(train_data.targets) == i)[0]
    np.random.shuffle(class_indices)
    class_subset_indices = class_indices[:1000]
    subset_indices_train.extend(class_subset_indices)

# Create samplers from these indices
sampler = SubsetRandomSampler(subset_indices_train)

train_loader = DataLoader(train_data, batch_size=32, sampler=sampler)
test_loader = DataLoader(test_data, batch_size=32)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
# Create an instance of the ResNet20 model
model = ResNet20()
model = model.to(device)

# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

def accuracy(model, data_loader):
    device = next(model.parameters()).device
    correct = 0
    total = 0
    with torch.no_grad():
        for images, labels in data_loader:
            images = images.to(device)
            labels = labels.to(device)
            outputs = model(images)
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()

    return 100 * correct / total ,total
```

Q1) Resnet model without Augmentation

```
train_accuracies = []
test_accuracies = []
losses = []
num_epochs=100
for epoch in range(num_epochs):
    model.train()
    epoch_loss = 0
    for i, data in enumerate(train_loader):
        inputs, labels = data
        inputs = inputs.to(device)
        labels = labels.to(device)
       # zero the parameter gradients
        optimizer.zero_grad()
       # forward pass
```

```
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        epoch_loss += loss.item()
        # backward pass and optimization
        loss.backward()
        optimizer.step()
    train_accuracy,total= accuracy(model, train_loader)
    test_accuracy,total = accuracy(model, test_loader)
    train_accuracies.append(train_accuracy)
    test_accuracies.append(test_accuracy)
    losses.append(epoch_loss/total)
    print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {epoch_loss/total},
Train Accuracy: {train_accuracy} %, Test Accuracy: {test_accuracy} %')

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 2)
plt.plot( train_accuracies, label='Train Accuracy')
plt.plot( test_accuracies, label='Test Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')



plt.subplot(1, 2, 1)
plt.plot(losses, label='Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
Epoch [1/100], Loss: 0.05397837452888489, Train Accuracy: 43.08 %,
Test Accuracy: 41.31 %
Epoch [2/100], Loss: 0.04470515347123146, Train Accuracy: 51.93 %,
Test Accuracy: 51.37 %
Epoch [3/100], Loss: 0.038785678398609164, Train Accuracy: 61.41 %,
Test Accuracy: 58.39 %
Epoch [4/100], Loss: 0.03418513247668743, Train Accuracy: 65.61 %,
Test Accuracy: 60.89 %
Epoch [5/100], Loss: 0.03078512607216835, Train Accuracy: 68.16 %,
Test Accuracy: 62.27 %
Epoch [6/100], Loss: 0.02786265659928322, Train Accuracy: 72.15 %,
Test Accuracy: 63.74 %
Epoch [7/100], Loss: 0.02491578665971756, Train Accuracy: 75.4 %, Test
Accuracy: 65.12 %
Epoch [8/100], Loss: 0.022765826699137688, Train Accuracy: 78.15 %,
Test Accuracy: 65.7 %
Epoch [9/100], Loss: 0.01994272277057171, Train Accuracy: 81.5 %, Test
Accuracy: 67.8 %
Epoch [10/100], Loss: 0.01719081103503704, Train Accuracy: 85.43 %,
Test Accuracy: 67.87 %
Epoch [11/100], Loss: 0.014603301659226417, Train Accuracy: 87.92 %,
Test Accuracy: 68.12 %
Epoch [12/100], Loss: 0.012386740328371525, Train Accuracy: 88.23 %,
Test Accuracy: 67.18 %
Epoch [13/100], Loss: 0.010537341342493891, Train Accuracy: 91.24 %,
Test Accuracy: 68.16 %
Epoch [14/100], Loss: 0.008633106910064817, Train Accuracy: 91.95 %,
Test Accuracy: 67.57 %
Epoch [15/100], Loss: 0.007686804256215691, Train Accuracy: 95.1 %,
Test Accuracy: 68.03 %
Epoch [16/100], Loss: 0.00662505198083818, Train Accuracy: 94.39 %,
Test Accuracy: 66.97 %
Epoch [17/100], Loss: 0.0053098703311756255, Train Accuracy: 94.87 %,
Test Accuracy: 67.16 %
Epoch [18/100], Loss: 0.00464136969409883, Train Accuracy: 96.46 %,
Test Accuracy: 67.09 %
Epoch [19/100], Loss: 0.004230955989100039, Train Accuracy: 95.21 %,
Test Accuracy: 65.64 %
Epoch [20/100], Loss: 0.003759392863512039, Train Accuracy: 96.07 %,
Test Accuracy: 66.31 %
Epoch [21/100], Loss: 0.0035161187984980645, Train Accuracy: 95.98 %,
Test Accuracy: 66.13 %
Epoch [22/100], Loss: 0.003793850875273347, Train Accuracy: 97.07 %,
Test Accuracy: 67.35 %
Epoch [23/100], Loss: 0.00318468445581384, Train Accuracy: 97.32 %,
Test Accuracy: 66.47 %
Epoch [24/100], Loss: 0.0033609208412468433, Train Accuracy: 97.56 %,
Test Accuracy: 66.94 %
Epoch [25/100], Loss: 0.0033112877128180116, Train Accuracy: 97.2 %,
Test Accuracy: 66.66 %
Epoch [26/100], Loss: 0.0028966105315368623, Train Accuracy: 98.02 %,
Test Accuracy: 67.67 %
Epoch [27/100], Loss: 0.0026462971259839834, Train Accuracy: 96.91 %,
Test Accuracy: 67.01 %
Epoch [28/100], Loss: 0.002465827270434238, Train Accuracy: 96.2 %,
Test Accuracy: 65.7 %
Epoch [29/100], Loss: 0.002438306281436235, Train Accuracy: 97.83 %,
Test Accuracy: 67.41 %
Epoch [30/100], Loss: 0.0018503837695578113, Train Accuracy: 98.93 %,
Test Accuracy: 67.77 %
Epoch [31/100], Loss: 0.0020722651300253345, Train Accuracy: 96.01 %,
Test Accuracy: 65.07 %
Epoch [32/100], Loss: 0.0031353330635931344, Train Accuracy: 98.36 %,
Test Accuracy: 67.59 %
Epoch [33/100], Loss: 0.0024045704340329393, Train Accuracy: 98.15 %,
Test Accuracy: 67.31 %
Epoch [34/100], Loss: 0.0018105296598980203, Train Accuracy: 98.49 %,
Test Accuracy: 67.89 %
Epoch [35/100], Loss: 0.0018236928292550147, Train Accuracy: 98.31 %,
Test Accuracy: 66.82 %
Epoch [36/100], Loss: 0.0022075779042905197, Train Accuracy: 98.86 %,
Test Accuracy: 67.3 %
Epoch [37/100], Loss: 0.0022911577450227924, Train Accuracy: 97.85 %,
Test Accuracy: 65.79 %
Epoch [38/100], Loss: 0.0020612002978567033, Train Accuracy: 98.32 %,
Test Accuracy: 67.31 %
Epoch [39/100], Loss: 0.001533562266489025, Train Accuracy: 98.79 %,
Test Accuracy: 67.74 %
Epoch [40/100], Loss: 0.0015934930227114819, Train Accuracy: 98.94 %,
Test Accuracy: 68.01 %
Epoch [41/100], Loss: 0.0013547916438983521, Train Accuracy: 98.6 %,
Test Accuracy: 67.53 %
Epoch [42/100], Loss: 0.0023216166886733847, Train Accuracy: 97.99 %,
Test Accuracy: 67.16 %
Epoch [43/100], Loss: 0.0020575697088032027, Train Accuracy: 98.56 %,
Test Accuracy: 67.45 %
Epoch [44/100], Loss: 0.0014241776046459563, Train Accuracy: 98.86 %,
Test Accuracy: 67.77 %
Epoch [45/100], Loss: 0.0014530869934911608, Train Accuracy: 98.89 %,
Test Accuracy: 67.59 %
Epoch [46/100], Loss: 0.001576554347248748, Train Accuracy: 98.25 %,
Test Accuracy: 67.49 %
Epoch [47/100], Loss: 0.0018715550499095116, Train Accuracy: 97.85 %,
Test Accuracy: 66.69 %
Epoch [48/100], Loss: 0.0015697146994643845, Train Accuracy: 98.64 %,
Test Accuracy: 67.45 %
Epoch [49/100], Loss: 0.001505794603159302, Train Accuracy: 98.26 %,
Test Accuracy: 66.53 %
Epoch [50/100], Loss: 0.001459662798550562, Train Accuracy: 98.25 %,
Test Accuracy: 66.88 %
Epoch [51/100], Loss: 0.001529961394140264, Train Accuracy: 98.67 %,
Test Accuracy: 67.19 %
Epoch [52/100], Loss: 0.0013362638898950536, Train Accuracy: 99.21 %,
Test Accuracy: 68.09 %
Epoch [53/100], Loss: 0.0012316098859359045, Train Accuracy: 98.6 %,
Test Accuracy: 66.86 %
Epoch [54/100], Loss: 0.0014509122102404944, Train Accuracy: 98.93 %,
Test Accuracy: 67.07 %
Epoch [55/100], Loss: 0.001162480658662389, Train Accuracy: 99.03 %,
Test Accuracy: 67.65 %
Epoch [56/100], Loss: 0.0011072113683185307, Train Accuracy: 99.18 %,
Test Accuracy: 67.52 %
Epoch [57/100], Loss: 0.0010459584084426751, Train Accuracy: 98.48 %,
Test Accuracy: 67.34 %
Epoch [58/100], Loss: 0.0019065156086697243, Train Accuracy: 98.82 %,
Test Accuracy: 67.01 %
Epoch [59/100], Loss: 0.0010761537676327862, Train Accuracy: 99.4 %,
Test Accuracy: 68.21 %
Epoch [60/100], Loss: 0.0007605619895693962, Train Accuracy: 99.2 %,
Test Accuracy: 67.57 %
Epoch [61/100], Loss: 0.0007939051211462356, Train Accuracy: 99.13 %,
Test Accuracy: 67.11 %
Epoch [62/100], Loss: 0.0013022255015472182, Train Accuracy: 98.5 %,
Test Accuracy: 66.54 %
Epoch [63/100], Loss: 0.0014445892111602007, Train Accuracy: 99.0 %,
Test Accuracy: 67.71 %
Epoch [64/100], Loss: 0.0009097355507401517, Train Accuracy: 99.12 %,
Test Accuracy: 67.25 %
Epoch [65/100], Loss: 0.0012337870175018906, Train Accuracy: 98.0 %,
Test Accuracy: 66.18 %
Epoch [66/100], Loss: 0.0011169947370071895, Train Accuracy: 99.09 %,
Test Accuracy: 68.06 %
Epoch [67/100], Loss: 0.001044756533158943, Train Accuracy: 99.14 %,
Test Accuracy: 67.04 %
Epoch [68/100], Loss: 0.001189137755810225, Train Accuracy: 98.68 %,
Test Accuracy: 66.28 %
Epoch [69/100], Loss: 0.0010688321792142234, Train Accuracy: 98.73 %,
Test Accuracy: 66.72 %
Epoch [70/100], Loss: 0.0012973653782246402, Train Accuracy: 98.83 %,
Test Accuracy: 67.42 %
Epoch [71/100], Loss: 0.0010192564145225333, Train Accuracy: 99.28 %,
Test Accuracy: 68.67 %
Epoch [72/100], Loss: 0.0007607354532425234, Train Accuracy: 99.26 %,
Test Accuracy: 67.43 %
Epoch [73/100], Loss: 0.000670909345036489, Train Accuracy: 99.64 %,
Test Accuracy: 67.88 %
Epoch [74/100], Loss: 0.0014033534172282088, Train Accuracy: 98.41 %,
Test Accuracy: 66.55 %
Epoch [75/100], Loss: 0.0008304226554813795, Train Accuracy: 99.61 %,
Test Accuracy: 68.23 %
Epoch [76/100], Loss: 0.000646583783897222, Train Accuracy: 98.91 %,
Test Accuracy: 67.11 %
Epoch [77/100], Loss: 0.0008976872807950713, Train Accuracy: 99.5 %,
Test Accuracy: 68.0 %
Epoch [78/100], Loss: 0.0011576962031875156, Train Accuracy: 99.25 %,
Test Accuracy: 67.52 %
Epoch [79/100], Loss: 0.0011355403093970381, Train Accuracy: 99.03 %,
Test Accuracy: 67.25 %
Epoch [80/100], Loss: 0.0008923379979154561, Train Accuracy: 99.15 %,
Test Accuracy: 67.06 %
Epoch [81/100], Loss: 0.0008827566276071593, Train Accuracy: 99.19 %,
Test Accuracy: 66.46 %
Epoch [82/100], Loss: 0.0010951966576540145, Train Accuracy: 98.67 %,
Test Accuracy: 65.23 %
Epoch [83/100], Loss: 0.0010520645059747039, Train Accuracy: 99.32 %,
Test Accuracy: 67.33 %
Epoch [84/100], Loss: 0.0007834875849133823, Train Accuracy: 99.25 %,
Test Accuracy: 67.63 %
Epoch [85/100], Loss: 0.0008796562379546231, Train Accuracy: 98.82 %,
Test Accuracy: 66.48 %
Epoch [86/100], Loss: 0.0010021262422669679, Train Accuracy: 99.09 %,
Test Accuracy: 67.0 %
Epoch [87/100], Loss: 0.0008342228628040175, Train Accuracy: 99.46 %,
Test Accuracy: 68.15 %
Epoch [88/100], Loss: 0.0003595888898991689, Train Accuracy: 99.48 %,
Test Accuracy: 67.59 %
Epoch [89/100], Loss: 0.0004612339926490677, Train Accuracy: 99.08 %,
Test Accuracy: 66.83 %
Epoch [90/100], Loss: 0.0006696545527374837, Train Accuracy: 99.03 %,
Test Accuracy: 67.3 %
Epoch [91/100], Loss: 0.001110676276765298, Train Accuracy: 99.1 %,
Test Accuracy: 67.13 %
Epoch [92/100], Loss: 0.0014494305370404618, Train Accuracy: 98.47 %,
Test Accuracy: 65.55 %
Epoch [93/100], Loss: 0.0009885447674896567, Train Accuracy: 99.33 %,
Test Accuracy: 67.73 %
Epoch [94/100], Loss: 0.0005659138017203077, Train Accuracy: 99.56 %,
Test Accuracy: 66.95 %
Epoch [95/100], Loss: 0.00046765019052036225, Train Accuracy: 99.67 %,
Test Accuracy: 67.5 %
Epoch [96/100], Loss: 0.000325900266753888, Train Accuracy: 99.72 %,
Test Accuracy: 67.48 %
Epoch [97/100], Loss: 0.00033306178761231424, Train Accuracy: 99.29 %,
Test Accuracy: 66.61 %
Epoch [98/100], Loss: 0.0014931818580749677, Train Accuracy: 98.57 %,
Test Accuracy: 66.19 %
Epoch [99/100], Loss: 0.0008985639453487238, Train Accuracy: 99.22 %,
Test Accuracy: 67.38 %
Epoch [100/100], Loss: 0.0006290891879121773, Train Accuracy: 99.22 %,
Test Accuracy: 67.68 %
```



```python
print('Test Accuracy for resnet20:', test_accuracy)
```

```
Test Accuracy for resnet20: 67.68
```

Q2) Mixup Augmentation

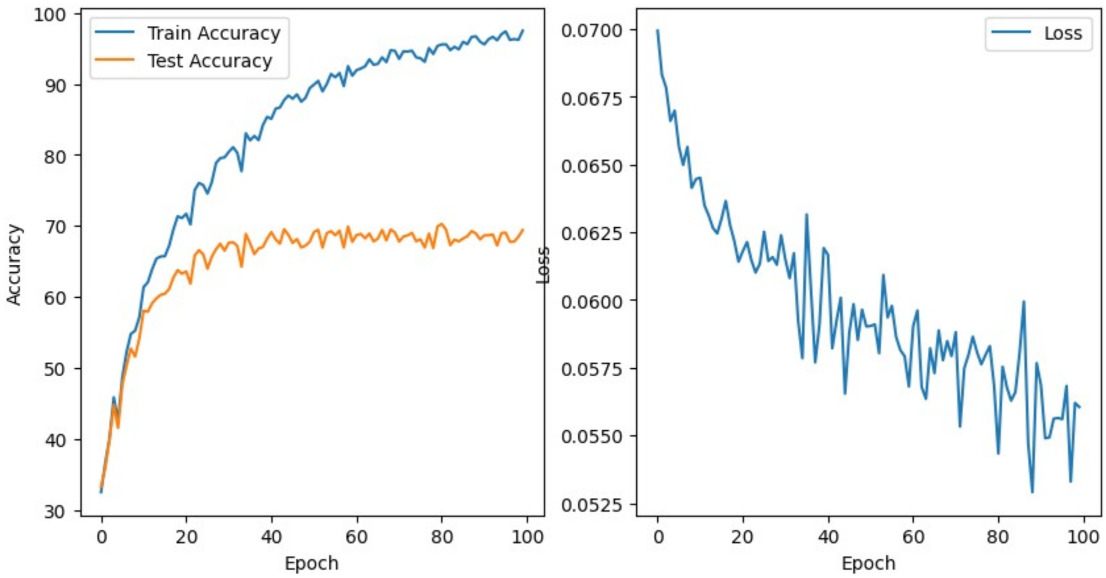
```
train_loader = DataLoader(train_data, batch_size=32, sampler=sampler)
test_loader = DataLoader(test_data, batch_size=32)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
# Create an instance of the ResNet20 model
model = ResNet20()
model = model.to(device)

# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

def mixup_data(x, y, alpha=1.0, device=device):
    '''Returns mixed inputs, pairs of targets, and lambda'''
    if alpha > 0:
        lam = np.random.beta(alpha, alpha)
    else:
        lam = 1

    batch_size = x.size()[0]
    index = torch.randperm(batch_size).to(device)
    index1= torch.randperm(batch_size).to(device)

    lam = torch.tensor(lam, device=device, dtype=x.dtype)
    x, y = x.to(device), y.to(device)
    s_indices = torch.randperm(batch_size).to(x.device)
    mixed_x = lam * x[index1,:] + (1 - lam) * x[index, :]
    mixed_y = lam.squeeze() * y + (1 - lam.squeeze()) * y[s_indices]
    y_a, y_b = mixed_y, y[index]
    return mixed_x, y_a, y_b, lam
```

Alpha: 0.2

```
num_epochs = 100
alpha = 0.2
train_accuracies = []
test_accuracies = []
losses = []
for epoch in range(num_epochs):
        model.train()
        epoch_loss=0
        for i, (images, labels) in enumerate(train_loader):
            images = images.to(device)
            labels = labels.to(device)
            # Apply mixup
            images, labels_a, labels_b, lam = mixup_data(images,
labels, alpha)
            labels_a = labels_a.long()
            labels_b = labels_b.long()
            # Forward pass
            outputs = model(images)
            loss = criterion(outputs, labels_a) * lam +
criterion(outputs, labels_b) * (1 - lam)
            # loss = criterion(outputs, labels_a)
            epoch_loss += loss.item()

            # Backward and optimize
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()

        train_accuracy,total = accuracy(model, train_loader)
        test_accuracy ,total= accuracy(model, test_loader)
        train_accuracies.append(train_accuracy)
        test_accuracies.append(test_accuracy)
        losses.append(epoch_loss/total)
        print (f'Epoch [{epoch+1}/{num_epochs}], Loss:
{epoch_loss/total}, Train Accuracy: {train_accuracy} %, Test Accuracy:
{test_accuracy} %')

plt.figure(figsize=(10,5))


plt.subplot(1, 2, 1)
plt.plot(train_accuracies, label='Train Accuracy')
plt.plot( test_accuracies, label='Test Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(losses, label='Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
```

```
Epoch [1/100], Loss: 0.06994095106124879, Train Accuracy: 32.47 %,
Test Accuracy: 33.22 %
Epoch [2/100], Loss: 0.0683220120549202, Train Accuracy: 36.56 %, Test
Accuracy: 35.93 %
Epoch [3/100], Loss: 0.06783041832447052, Train Accuracy: 40.06 %,
Test Accuracy: 39.92 %
Epoch [4/100], Loss: 0.0666094445347786, Train Accuracy: 45.83 %, Test
Accuracy: 44.77 %
Epoch [5/100], Loss: 0.06699267574548722, Train Accuracy: 42.46 %,
Test Accuracy: 41.52 %
Epoch [6/100], Loss: 0.06566099425554275, Train Accuracy: 48.63 %,
Test Accuracy: 47.64 %
Epoch [7/100], Loss: 0.06498629641532898, Train Accuracy: 52.21 %,
Test Accuracy: 50.31 %
Epoch [8/100], Loss: 0.0656491003036499, Train Accuracy: 54.77 %, Test
Accuracy: 52.71 %
Epoch [9/100], Loss: 0.06413923705816268, Train Accuracy: 55.22 %,
Test Accuracy: 51.57 %
Epoch [10/100], Loss: 0.06446351972818375, Train Accuracy: 57.13 %,
Test Accuracy: 54.15 %
Epoch [11/100], Loss: 0.06451433844566345, Train Accuracy: 61.37 %,
Test Accuracy: 58.04 %
Epoch [12/100], Loss: 0.06349557659626007, Train Accuracy: 62.05 %,
Test Accuracy: 57.89 %
Epoch [13/100], Loss: 0.06311909215450287, Train Accuracy: 63.88 %,
Test Accuracy: 59.1 %
Epoch [14/100], Loss: 0.06265741444826126, Train Accuracy: 65.37 %,
Test Accuracy: 59.77 %
Epoch [15/100], Loss: 0.06245432305335998, Train Accuracy: 65.68 %,
Test Accuracy: 60.28 %
Epoch [16/100], Loss: 0.06299597851037979, Train Accuracy: 65.72 %,
Test Accuracy: 60.43 %
Epoch [17/100], Loss: 0.06365490583181381, Train Accuracy: 67.2 %,
Test Accuracy: 61.1 %
Epoch [18/100], Loss: 0.06277075204849243, Train Accuracy: 69.5 %,
Test Accuracy: 62.72 %
Epoch [19/100], Loss: 0.062178983056545255, Train Accuracy: 71.34 %,
Test Accuracy: 63.76 %
Epoch [20/100], Loss: 0.061413909995555876, Train Accuracy: 71.09 %,
Test Accuracy: 63.26 %
Epoch [21/100], Loss: 0.061800358140468595, Train Accuracy: 71.7 %,
Test Accuracy: 63.58 %
Epoch [22/100], Loss: 0.062138995152711866, Train Accuracy: 70.19 %,
Test Accuracy: 61.84 %
Epoch [23/100], Loss: 0.061478380119800564, Train Accuracy: 75.04 %,
Test Accuracy: 65.82 %
Epoch [24/100], Loss: 0.0610141615152359, Train Accuracy: 76.04 %,
Test Accuracy: 66.57 %
Epoch [25/100], Loss: 0.06133948577046394, Train Accuracy: 75.74 %,
Test Accuracy: 65.98 %
Epoch [26/100], Loss: 0.06252213332057, Train Accuracy: 74.54 %, Test
Accuracy: 63.95 %
Epoch [27/100], Loss: 0.06144046870470047, Train Accuracy: 76.14 %,
Test Accuracy: 65.63 %
Epoch [28/100], Loss: 0.06158410922288895, Train Accuracy: 78.84 %,
Test Accuracy: 66.7 %
Epoch [29/100], Loss: 0.061302500355243685, Train Accuracy: 79.54 %,
Test Accuracy: 67.46 %
Epoch [30/100], Loss: 0.06238865023255348, Train Accuracy: 79.64 %,
Test Accuracy: 66.5 %
Epoch [31/100], Loss: 0.061490587741136554, Train Accuracy: 80.42 %,
Test Accuracy: 67.62 %
Epoch [32/100], Loss: 0.060806887525320055, Train Accuracy: 81.08 %,
Test Accuracy: 67.69 %
Epoch [33/100], Loss: 0.06172505888938904, Train Accuracy: 80.28 %,
Test Accuracy: 67.17 %
Epoch [34/100], Loss: 0.059234687054157256, Train Accuracy: 77.69 %,
Test Accuracy: 64.23 %
Epoch [35/100], Loss: 0.05785162037014961, Train Accuracy: 83.03 %,
Test Accuracy: 68.85 %
Epoch [36/100], Loss: 0.06315827401280404, Train Accuracy: 82.03 %,
Test Accuracy: 67.52 %
Epoch [37/100], Loss: 0.060374756002426144, Train Accuracy: 82.66 %,
Test Accuracy: 66.01 %
Epoch [38/100], Loss: 0.05769822179675102, Train Accuracy: 82.07 %,
Test Accuracy: 66.81 %
Epoch [39/100], Loss: 0.059087744623422624, Train Accuracy: 84.19 %,
Test Accuracy: 67.02 %
Epoch [40/100], Loss: 0.061918244737386705, Train Accuracy: 85.35 %,
Test Accuracy: 68.19 %
Epoch [41/100], Loss: 0.06166446349620819, Train Accuracy: 85.07 %,
Test Accuracy: 69.11 %
Epoch [42/100], Loss: 0.058218680888414386, Train Accuracy: 86.51 %,
Test Accuracy: 68.08 %
Epoch [43/100], Loss: 0.05922876070737839, Train Accuracy: 86.66 %,
Test Accuracy: 67.48 %
Epoch [44/100], Loss: 0.060083469361066816, Train Accuracy: 87.72 %,
Test Accuracy: 69.52 %
Epoch [45/100], Loss: 0.056542472130060195, Train Accuracy: 88.36 %,
Test Accuracy: 68.72 %
Epoch [46/100], Loss: 0.0587885118663311, Train Accuracy: 87.91 %,
Test Accuracy: 67.59 %
Epoch [47/100], Loss: 0.05984924133419991, Train Accuracy: 88.52 %,
Test Accuracy: 68.14 %
Epoch [48/100], Loss: 0.05852396418452263, Train Accuracy: 87.49 %,
Test Accuracy: 66.95 %
Epoch [49/100], Loss: 0.059643139123916625, Train Accuracy: 88.02 %,
Test Accuracy: 67.2 %
Epoch [50/100], Loss: 0.05902410826086998, Train Accuracy: 89.39 %,
Test Accuracy: 67.75 %
Epoch [51/100], Loss: 0.05904164603948593, Train Accuracy: 89.9 %,
Test Accuracy: 69.11 %
Epoch [52/100], Loss: 0.05910091664195061, Train Accuracy: 90.42 %,
Test Accuracy: 69.45 %
Epoch [53/100], Loss: 0.05803695895075798, Train Accuracy: 88.94 %,
Test Accuracy: 66.93 %
Epoch [54/100], Loss: 0.060928284060955046, Train Accuracy: 89.97 %,
Test Accuracy: 68.95 %
Epoch [55/100], Loss: 0.059354704177379605, Train Accuracy: 91.39 %,
Test Accuracy: 69.25 %
Epoch [56/100], Loss: 0.05978314730525017, Train Accuracy: 90.93 %,
Test Accuracy: 68.67 %
Epoch [57/100], Loss: 0.05863459888100624, Train Accuracy: 91.53 %,
Test Accuracy: 69.31 %
Epoch [58/100], Loss: 0.05814767074584961, Train Accuracy: 89.69 %,
Test Accuracy: 66.94 %
Epoch [59/100], Loss: 0.05792381809949875, Train Accuracy: 92.5 %,
Test Accuracy: 69.91 %
Epoch [60/100], Loss: 0.05680913099050522, Train Accuracy: 91.15 %,
Test Accuracy: 67.74 %
Epoch [61/100], Loss: 0.059028378456830975, Train Accuracy: 91.94 %,
Test Accuracy: 68.71 %
Epoch [62/100], Loss: 0.05961068931818008, Train Accuracy: 92.16 %,
Test Accuracy: 68.86 %
Epoch [63/100], Loss: 0.05678659321069717, Train Accuracy: 92.47 %,
Test Accuracy: 68.23 %
Epoch [64/100], Loss: 0.056349621510505675, Train Accuracy: 93.45 %,
Test Accuracy: 68.92 %
Epoch [65/100], Loss: 0.058215749657154085, Train Accuracy: 92.7 %,
Test Accuracy: 67.8 %
Epoch [66/100], Loss: 0.0573023598074913, Train Accuracy: 92.81 %,
Test Accuracy: 68.2 %
Epoch [67/100], Loss: 0.05887762022614479, Train Accuracy: 93.74 %,
Test Accuracy: 69.42 %
Epoch [68/100], Loss: 0.05777505051493645, Train Accuracy: 93.07 %,
Test Accuracy: 67.95 %
Epoch [69/100], Loss: 0.058486318492889405, Train Accuracy: 94.73 %,
Test Accuracy: 69.48 %
Epoch [70/100], Loss: 0.05793020572662354, Train Accuracy: 94.66 %,
Test Accuracy: 69.06 %
Epoch [71/100], Loss: 0.05881510224342346, Train Accuracy: 93.52 %,
Test Accuracy: 67.78 %
Epoch [72/100], Loss: 0.05532902464270592, Train Accuracy: 94.58 %,
Test Accuracy: 68.49 %
Epoch [73/100], Loss: 0.05748422635793686, Train Accuracy: 94.55 %,
Test Accuracy: 68.65 %
Epoch [74/100], Loss: 0.05798097338676453, Train Accuracy: 94.68 %,
Test Accuracy: 69.01 %
Epoch [75/100], Loss: 0.058646100914478305, Train Accuracy: 93.73 %,
Test Accuracy: 67.82 %
Epoch [76/100], Loss: 0.058068327069282534, Train Accuracy: 93.61 %,
Test Accuracy: 68.1 %
Epoch [77/100], Loss: 0.0576274973154068, Train Accuracy: 93.11 %,
Test Accuracy: 66.93 %
Epoch [78/100], Loss: 0.05797620945572853, Train Accuracy: 95.05 %,
Test Accuracy: 68.91 %
Epoch [79/100], Loss: 0.05829939151406288, Train Accuracy: 94.21 %,
Test Accuracy: 66.87 %
Epoch [80/100], Loss: 0.056842392152547834, Train Accuracy: 95.35 %,
Test Accuracy: 69.91 %
Epoch [81/100], Loss: 0.05433120764493942, Train Accuracy: 95.55 %,
Test Accuracy: 70.26 %
Epoch [82/100], Loss: 0.05753563714027405, Train Accuracy: 95.57 %,
Test Accuracy: 69.46 %
Epoch [83/100], Loss: 0.05677750238776207, Train Accuracy: 94.75 %,
Test Accuracy: 67.27 %
Epoch [84/100], Loss: 0.05628428260684013, Train Accuracy: 95.23 %,
Test Accuracy: 68.05 %
Epoch [85/100], Loss: 0.05660165485739708, Train Accuracy: 94.86 %,
Test Accuracy: 67.8 %
Epoch [86/100], Loss: 0.05806820773482323, Train Accuracy: 95.93 %,
Test Accuracy: 68.2 %
Epoch [87/100], Loss: 0.05994302330613136, Train Accuracy: 95.59 %,
Test Accuracy: 68.53 %
Epoch [88/100], Loss: 0.05472601215839386, Train Accuracy: 96.61 %,
Test Accuracy: 69.26 %
Epoch [89/100], Loss: 0.052906014078855514, Train Accuracy: 96.7 %,
Test Accuracy: 68.93 %
Epoch [90/100], Loss: 0.05767038713693619, Train Accuracy: 95.94 %,
Test Accuracy: 68.09 %
Epoch [91/100], Loss: 0.0568545376598835, Train Accuracy: 95.56 %,
Test Accuracy: 68.66 %
Epoch [92/100], Loss: 0.054896426266431805, Train Accuracy: 96.29 %,
Test Accuracy: 68.67 %
Epoch [93/100], Loss: 0.05493006276488304, Train Accuracy: 96.63 %,
Test Accuracy: 68.77 %
Epoch [94/100], Loss: 0.055618669641017915, Train Accuracy: 96.14 %,
Test Accuracy: 67.23 %
Epoch [95/100], Loss: 0.05564472103714943, Train Accuracy: 96.99 %,
Test Accuracy: 68.96 %
Epoch [96/100], Loss: 0.05560122554302216, Train Accuracy: 97.4 %,
Test Accuracy: 69.04 %
Epoch [97/100], Loss: 0.056828197729587554, Train Accuracy: 96.19 %,
Test Accuracy: 67.75 %
Epoch [98/100], Loss: 0.0532971533626318, Train Accuracy: 96.32 %,
Test Accuracy: 67.77 %
Epoch [99/100], Loss: 0.05620322485566139, Train Accuracy: 96.2 %,
Test Accuracy: 68.45 %
Epoch [100/100], Loss: 0.056049891567230226, Train Accuracy: 97.49 %,
Test Accuracy: 69.41 %
```

```
<matplotlib.legend.Legend at 0x7faa98c26800>
```



```
print('Test Accuracy for Mixup with Alpha=0.2:', test_accuracy)
```

```
Test Accuracy for Mixup with Alpha=0.2: 69.41
```

```
train_loader = DataLoader(train_data, batch_size=32, sampler=sampler)
test_loader = DataLoader(test_data, batch_size=32)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
# Create an instance of the ResNet20 model
model = ResNet20()
model = model.to(device)  # move the model to GPU if available

# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

Alpha: 0.4

```
num_epochs = 100
alpha = 0.4
train_accuracies = []
test_accuracies = []
losses = []
for epoch in range(num_epochs):
        model.train()
        epoch_loss=0
        for i, (images, labels) in enumerate(train_loader):
            images = images.to(device)
            labels = labels.to(device)
            # Apply mixup
            images, labels_a, labels_b, lam = mixup_data(images,
labels, alpha)
            labels_a = labels_a.long()
            labels_b = labels_b.long()
            # Forward pass
            outputs = model(images)
            loss = criterion(outputs, labels_a) * lam +
criterion(outputs, labels_b) * (1 - lam)
            # loss = criterion(outputs, labels_a)
            epoch_loss += loss.item()
            # Backward and optimize
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()

        train_accuracy,total = accuracy(model, train_loader)
        test_accuracy ,total= accuracy(model, test_loader)
        train_accuracies.append(train_accuracy)
        test_accuracies.append(test_accuracy)

        losses.append(epoch_loss/total)

        print (f'Epoch [{epoch+1}/{num_epochs}], Loss:
{epoch_loss/total}, Train Accuracy: {train_accuracy} %, Test Accuracy:
{test_accuracy} %')

plt.figure(figsize=(10,5))


plt.subplot(1, 2, 1)
plt.plot(train_accuracies, label='Train Accuracy')
plt.plot( test_accuracies, label='Test Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(losses, label='Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
```

```
Epoch [1/100], Loss: 0.07049459052085877, Train Accuracy: 28.17 %,
Test Accuracy: 27.45 %
Epoch [2/100], Loss: 0.06908390514850617, Train Accuracy: 33.74 %,
Test Accuracy: 33.11 %
Epoch [3/100], Loss: 0.06802033567428589, Train Accuracy: 41.14 %,
Test Accuracy: 40.57 %
Epoch [4/100], Loss: 0.0683674599647522, Train Accuracy: 40.1 %, Test
Accuracy: 39.73 %
Epoch [5/100], Loss: 0.06817946363687516, Train Accuracy: 42.47 %,
Test Accuracy: 42.1 %
Epoch [6/100], Loss: 0.06717940397262573, Train Accuracy: 45.72 %,
Test Accuracy: 45.42 %
```

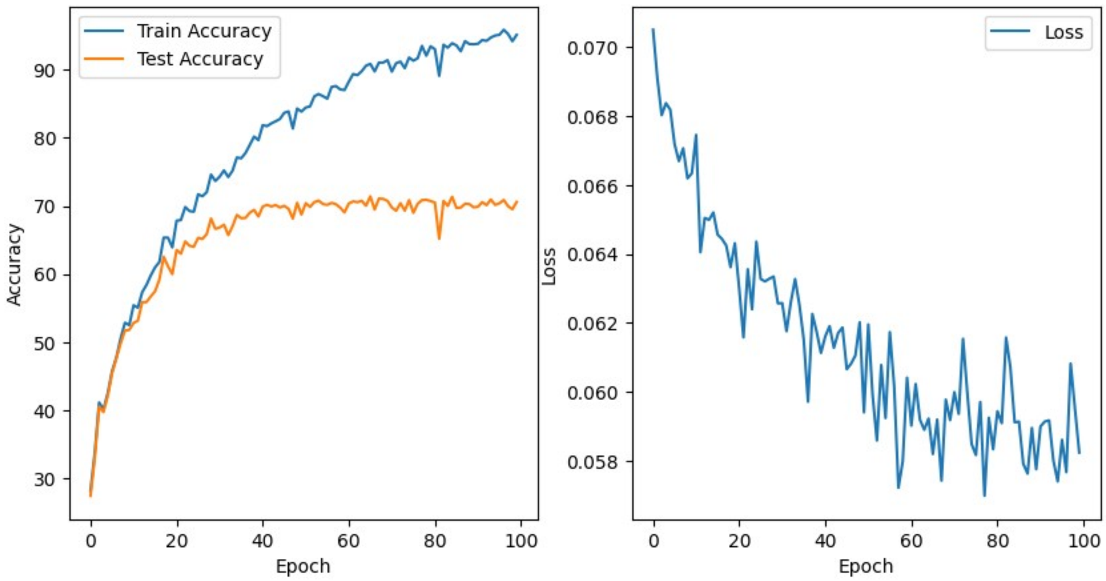
```
Epoch [7/100], Loss: 0.06668954546451569, Train Accuracy: 47.72 %,
Test Accuracy: 47.66 %
Epoch [8/100], Loss: 0.06706147311925888, Train Accuracy: 50.54 %,
Test Accuracy: 49.83 %
Epoch [9/100], Loss: 0.0661894180774688, Train Accuracy: 52.85 %,
Test Accuracy: 51.7 %
Epoch [10/100], Loss: 0.0663345610499382, Train Accuracy: 52.5 %, Test
Accuracy: 51.79 %
Epoch [11/100], Loss: 0.06744869068861008, Train Accuracy: 55.41 %,
Test Accuracy: 52.82 %
Epoch [12/100], Loss: 0.0640508123755455, Train Accuracy: 55.05 %,
Test Accuracy: 53.15 %
Epoch [13/100], Loss: 0.06504353507757187, Train Accuracy: 57.33 %,
Test Accuracy: 55.86 %
Epoch [14/100], Loss: 0.06498813877105714, Train Accuracy: 58.4 %,
Test Accuracy: 55.88 %
Epoch [15/100], Loss: 0.06520157059431075, Train Accuracy: 59.8 %,
Test Accuracy: 56.72 %
Epoch [16/100], Loss: 0.06455985388755799, Train Accuracy: 60.98 %,
Test Accuracy: 57.47 %
Epoch [17/100], Loss: 0.06443645174503326, Train Accuracy: 61.77 %,
Test Accuracy: 59.18 %
Epoch [18/100], Loss: 0.06424387985467911, Train Accuracy: 65.35 %,
Test Accuracy: 62.53 %
Epoch [19/100], Loss: 0.0636175879240036, Train Accuracy: 65.34 %,
Test Accuracy: 61.13 %
Epoch [20/100], Loss: 0.0643099344611168, Train Accuracy: 63.91 %,
Test Accuracy: 59.99 %
Epoch [21/100], Loss: 0.0630124344587326, Train Accuracy: 67.82 %,
Test Accuracy: 63.52 %
Epoch [22/100], Loss: 0.06158411191701889, Train Accuracy: 67.93 %,
Test Accuracy: 62.98 %
Epoch [23/100], Loss: 0.06355800703763961, Train Accuracy: 69.83 %,
Test Accuracy: 64.78 %
Epoch [24/100], Loss: 0.06239587714672089, Train Accuracy: 69.24 %,
Test Accuracy: 64.14 %
Epoch [25/100], Loss: 0.06435604392290116, Train Accuracy: 69.15 %,
Test Accuracy: 64.01 %
Epoch [26/100], Loss: 0.0632754925251007, Train Accuracy: 71.7 %, Test
Accuracy: 65.31 %
Epoch [27/100], Loss: 0.06320556002855302, Train Accuracy: 71.42 %,
Test Accuracy: 65.15 %
Epoch [28/100], Loss: 0.06328174599409103, Train Accuracy: 72.02 %,
Test Accuracy: 65.87 %
Epoch [29/100], Loss: 0.06334619390964508, Train Accuracy: 74.59 %,
Test Accuracy: 68.14 %
Epoch [30/100], Loss: 0.06256697161197662, Train Accuracy: 73.66 %,
Test Accuracy: 66.63 %
Epoch [31/100], Loss: 0.06257635861635208, Train Accuracy: 74.29 %,
Test Accuracy: 66.83 %
```

```
Epoch [32/100], Loss: 0.06176377283334732, Train Accuracy: 75.21 %,
Test Accuracy: 67.24 %
Epoch [33/100], Loss: 0.06260046133995056, Train Accuracy: 74.24 %,
Test Accuracy: 65.72 %
Epoch [34/100], Loss: 0.0632742467045784, Train Accuracy: 75.18 %,
Test Accuracy: 67.03 %
Epoch [35/100], Loss: 0.06252373299598694, Train Accuracy: 77.11 %,
Test Accuracy: 68.68 %
Epoch [36/100], Loss: 0.06152289308309555, Train Accuracy: 76.99 %,
Test Accuracy: 68.19 %
Epoch [37/100], Loss: 0.05971979202628136, Train Accuracy: 77.76 %,
Test Accuracy: 68.24 %
Epoch [38/100], Loss: 0.0622619479060173, Train Accuracy: 78.96 %,
Test Accuracy: 69.0 %
Epoch [39/100], Loss: 0.06174353438019752, Train Accuracy: 80.18 %,
Test Accuracy: 69.41 %
Epoch [40/100], Loss: 0.06113298777937889, Train Accuracy: 79.65 %,
Test Accuracy: 68.44 %
Epoch [41/100], Loss: 0.06161157511472702, Train Accuracy: 81.85 %,
Test Accuracy: 69.9 %
Epoch [42/100], Loss: 0.061908913612365724, Train Accuracy: 81.71 %,
Test Accuracy: 70.16 %
Epoch [43/100], Loss: 0.06128103622794151, Train Accuracy: 82.11 %,
Test Accuracy: 69.92 %
Epoch [44/100], Loss: 0.061711413317918774, Train Accuracy: 82.42 %,
Test Accuracy: 70.12 %
Epoch [45/100], Loss: 0.061869719845056535, Train Accuracy: 82.76 %,
Test Accuracy: 69.77 %
Epoch [46/100], Loss: 0.06066305341720581, Train Accuracy: 83.66 %,
Test Accuracy: 70.01 %
Epoch [47/100], Loss: 0.06082182068824768, Train Accuracy: 83.87 %,
Test Accuracy: 69.58 %
Epoch [48/100], Loss: 0.06106218891143799, Train Accuracy: 81.36 %,
Test Accuracy: 68.14 %
Epoch [49/100], Loss: 0.06201957259774208, Train Accuracy: 84.29 %,
Test Accuracy: 70.45 %
Epoch [50/100], Loss: 0.05941213264465332, Train Accuracy: 83.84 %,
Test Accuracy: 68.73 %
Epoch [51/100], Loss: 0.06195899893641472, Train Accuracy: 84.44 %,
Test Accuracy: 70.4 %
Epoch [52/100], Loss: 0.05990723373293877, Train Accuracy: 84.6 %,
Test Accuracy: 69.85 %
Epoch [53/100], Loss: 0.05859670526385307, Train Accuracy: 86.1 %,
Test Accuracy: 70.56 %
Epoch [54/100], Loss: 0.060782447618246076, Train Accuracy: 86.42 %,
Test Accuracy: 70.77 %
Epoch [55/100], Loss: 0.05924822030067444, Train Accuracy: 86.1 %,
Test Accuracy: 70.27 %
Epoch [56/100], Loss: 0.06173608302474022, Train Accuracy: 85.74 %,
Test Accuracy: 70.16 %
```

```
Epoch [57/100], Loss: 0.06018932906985283, Train Accuracy: 87.46 %,
Test Accuracy: 70.44 %
Epoch [58/100], Loss: 0.05722809913754463, Train Accuracy: 87.59 %,
Test Accuracy: 70.25 %
Epoch [59/100], Loss: 0.05799490332603455, Train Accuracy: 87.1 %,
Test Accuracy: 69.74 %
Epoch [60/100], Loss: 0.06041595577001572, Train Accuracy: 87.01 %,
Test Accuracy: 69.05 %
Epoch [61/100], Loss: 0.05902760829925537, Train Accuracy: 88.23 %,
Test Accuracy: 70.35 %
Epoch [62/100], Loss: 0.060233088505268095, Train Accuracy: 89.36 %,
Test Accuracy: 70.67 %
Epoch [63/100], Loss: 0.05920131087899208, Train Accuracy: 89.2 %,
Test Accuracy: 70.55 %
Epoch [64/100], Loss: 0.058906809002161026, Train Accuracy: 89.77 %,
Test Accuracy: 70.75 %
Epoch [65/100], Loss: 0.05923376319408417, Train Accuracy: 90.58 %,
Test Accuracy: 70.02 %
Epoch [66/100], Loss: 0.058205288875103, Train Accuracy: 90.85 %, Test
Accuracy: 71.4 %
Epoch [67/100], Loss: 0.059203090518712996, Train Accuracy: 89.74 %,
Test Accuracy: 69.47 %
Epoch [68/100], Loss: 0.05743375859260559, Train Accuracy: 91.04 %,
Test Accuracy: 71.08 %
Epoch [69/100], Loss: 0.059780578589439394, Train Accuracy: 91.04 %,
Test Accuracy: 71.02 %
Epoch [70/100], Loss: 0.05918788692355156, Train Accuracy: 91.4 %,
Test Accuracy: 70.73 %
Epoch [71/100], Loss: 0.05999813708662987, Train Accuracy: 89.71 %,
Test Accuracy: 69.76 %
Epoch [72/100], Loss: 0.059369119822978975, Train Accuracy: 90.95 %,
Test Accuracy: 69.29 %
Epoch [73/100], Loss: 0.061542294883728026, Train Accuracy: 91.18 %,
Test Accuracy: 70.4 %
Epoch [74/100], Loss: 0.05997411978840828, Train Accuracy: 90.22 %,
Test Accuracy: 69.3 %
Epoch [75/100], Loss: 0.05849304672479629, Train Accuracy: 91.78 %,
Test Accuracy: 70.84 %
Epoch [76/100], Loss: 0.058175290703773495, Train Accuracy: 91.32 %,
Test Accuracy: 68.99 %
Epoch [77/100], Loss: 0.059711097007989884, Train Accuracy: 91.67 %,
Test Accuracy: 70.31 %
Epoch [78/100], Loss: 0.05699650844931602, Train Accuracy: 93.49 %,
Test Accuracy: 70.85 %
Epoch [79/100], Loss: 0.05925675781369209, Train Accuracy: 92.05 %,
Test Accuracy: 70.9 %
Epoch [80/100], Loss: 0.05834275413751602, Train Accuracy: 93.43 %,
Test Accuracy: 70.71 %
Epoch [81/100], Loss: 0.059445191037654874, Train Accuracy: 93.0 %,
Test Accuracy: 70.49 %
```

```
Epoch [82/100], Loss: 0.05909828872680664, Train Accuracy: 89.09 %,
Test Accuracy: 65.18 %
Epoch [83/100], Loss: 0.061578077185153964, Train Accuracy: 93.65 %,
Test Accuracy: 70.72 %
Epoch [84/100], Loss: 0.06071887210607529, Train Accuracy: 93.23 %,
Test Accuracy: 70.0 %
Epoch [85/100], Loss: 0.05912488520145416, Train Accuracy: 93.88 %,
Test Accuracy: 71.34 %
Epoch [86/100], Loss: 0.059143943548202516, Train Accuracy: 93.56 %,
Test Accuracy: 69.71 %
Epoch [87/100], Loss: 0.057922264522314075, Train Accuracy: 92.7 %,
Test Accuracy: 69.75 %
Epoch [88/100], Loss: 0.05763749381899834, Train Accuracy: 94.18 %,
Test Accuracy: 70.31 %
Epoch [89/100], Loss: 0.05896064655780792, Train Accuracy: 93.76 %,
Test Accuracy: 70.27 %
Epoch [90/100], Loss: 0.05776518918275833, Train Accuracy: 93.73 %,
Test Accuracy: 69.79 %
Epoch [91/100], Loss: 0.05900497016310692, Train Accuracy: 93.8 %,
Test Accuracy: 69.88 %
Epoch [92/100], Loss: 0.05914620844721794, Train Accuracy: 94.35 %,
Test Accuracy: 70.55 %
Epoch [93/100], Loss: 0.0591800173163414, Train Accuracy: 94.23 %,
Test Accuracy: 70.11 %
Epoch [94/100], Loss: 0.05801413691043854, Train Accuracy: 94.72 %,
Test Accuracy: 70.94 %
Epoch [95/100], Loss: 0.05741082049608231, Train Accuracy: 95.01 %,
Test Accuracy: 70.15 %
Epoch [96/100], Loss: 0.05861744567155838, Train Accuracy: 95.12 %,
Test Accuracy: 70.39 %
Epoch [97/100], Loss: 0.05768146266341209, Train Accuracy: 95.86 %,
Test Accuracy: 70.87 %
Epoch [98/100], Loss: 0.06082433099746704, Train Accuracy: 95.23 %,
Test Accuracy: 69.96 %
Epoch [99/100], Loss: 0.05952899745702744, Train Accuracy: 94.18 %,
Test Accuracy: 69.51 %
Epoch [100/100], Loss: 0.0582468396961689, Train Accuracy: 95.11 %,
Test Accuracy: 70.58 %
```

```
<matplotlib.legend.Legend at 0x7faaa9e78a00>
```



```
print('Test Accuracy for Mixup with Alpha=0.4:', test_accuracy)
```

```
Test Accuracy for Mixup with Alpha=0.4: 70.58
```

Cutout Augmentation

```
train_loader = DataLoader(train_data, batch_size=32, sampler=sampler)
test_loader = DataLoader(test_data, batch_size=32)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
# Create an instance of the ResNet20 model
model = ResNet20()
model = model.to(device)  # move the model to GPU if available

# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

class Cutout(object):
    def __init__(self, n_holes, length):
        self.n_holes = n_holes
        self.length = length

    def __call__(self, img):
        h, w = img.shape[-2], img.shape[-1]
        for n in range(self.n_holes):
            y = torch.randint(h, (1,))
            x = torch.randint(w, (1,))

            y1 = torch.clamp(y - self.length // 2, 0, h)
            y2 = torch.clamp(y + self.length // 2, 0, h)
            x1 = torch.clamp(x - self.length // 2, 0, w)
            x2 = torch.clamp(x + self.length // 2, 0, w)
            img[..., y1:y2, x1:x2] = 0.

        return img

cutout = Cutout(n_holes=1, length=16)


train_accuracies = []
test_accuracies = []
losses = []
num_epochs=100
for epoch in range(num_epochs):
    model.train()
    epoch_loss = 0
    for i, data in enumerate(train_loader):

        images, labels = data
        images=cutout(images)
        images = images.to(device)
        labels = labels.to(device)
     # zero the parameter gradients
        optimizer.zero_grad()

        # forward pass
        outputs = model(images)
        loss = criterion(outputs, labels)
        epoch_loss += loss.item()
        # backward pass and optimization
        loss.backward()
        optimizer.step()

    train_accuracy,total = accuracy(model, train_loader)
    test_accuracy ,total= accuracy(model, test_loader)
    train_accuracies.append(train_accuracy)
    test_accuracies.append(test_accuracy)

    losses.append(epoch_loss / total)

    print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {epoch_loss /
total}, Train Accuracy: {train_accuracy} %, Test Accuracy:
{test_accuracy} %')



plt.figure(figsize=(10,5))
plt.subplot(1, 2, 1)
plt.plot(losses)
plt.title('Training Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')


plt.subplot(1, 2, 2)
plt.plot(train_accuracies, label='Training Accuracy')
plt.plot(test_accuracies, label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

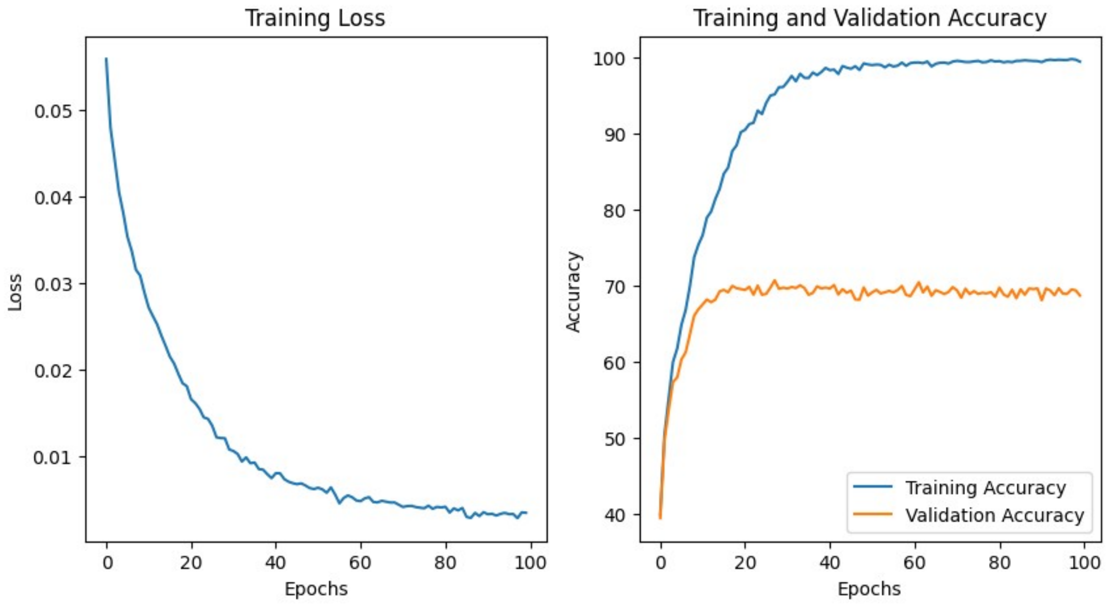
```
Epoch [1/100], Loss: 0.055918360364437106, Train Accuracy: 39.92 %,
Test Accuracy: 39.46 %
Epoch [2/100], Loss: 0.047995667767524716, Train Accuracy: 50.36 %,
Test Accuracy: 49.69 %
Epoch [3/100], Loss: 0.04426509531736374, Train Accuracy: 55.21 %,
Test Accuracy: 53.71 %
Epoch [4/100], Loss: 0.04053893908262253, Train Accuracy: 59.95 %,
Test Accuracy: 57.33 %
Epoch [5/100], Loss: 0.038166742980480194, Train Accuracy: 61.73 %,
Test Accuracy: 57.94 %
Epoch [6/100], Loss: 0.03536574345827102, Train Accuracy: 64.84 %,
Test Accuracy: 60.29 %
Epoch [7/100], Loss: 0.03375727555155754, Train Accuracy: 66.79 %,
Test Accuracy: 61.24 %
Epoch [8/100], Loss: 0.03156480892896652, Train Accuracy: 69.94 %,
Test Accuracy: 63.61 %
Epoch [9/100], Loss: 0.030880820530653, Train Accuracy: 73.72 %, Test
Accuracy: 66.04 %
Epoch [10/100], Loss: 0.028926769602298737, Train Accuracy: 75.36 %,
Test Accuracy: 66.88 %
Epoch [11/100], Loss: 0.027199140670895575, Train Accuracy: 76.6 %,
Test Accuracy: 67.51 %
Epoch [12/100], Loss: 0.02620519362092018, Train Accuracy: 78.91 %,
Test Accuracy: 68.16 %
Epoch [13/100], Loss: 0.0252527018815279, Train Accuracy: 79.71 %,
Test Accuracy: 67.81 %
Epoch [14/100], Loss: 0.023943961971998214, Train Accuracy: 81.36 %,
Test Accuracy: 68.15 %
Epoch [15/100], Loss: 0.022757706767320633, Train Accuracy: 82.68 %,
Test Accuracy: 69.22 %
Epoch [16/100], Loss: 0.021515740478038788, Train Accuracy: 84.68 %,
Test Accuracy: 69.43 %
Epoch [17/100], Loss: 0.020721314841508864, Train Accuracy: 85.47 %,
Test Accuracy: 69.11 %
Epoch [18/100], Loss: 0.019535786986351014, Train Accuracy: 87.67 %,
Test Accuracy: 69.95 %
Epoch [19/100], Loss: 0.01844199022203684, Train Accuracy: 88.39 %,
Test Accuracy: 69.65 %
```

```
Epoch [20/100], Loss: 0.018095430234074592, Train Accuracy: 90.13 %,
Test Accuracy: 69.54 %
Epoch [21/100], Loss: 0.016595129217207433, Train Accuracy: 90.43 %,
Test Accuracy: 69.41 %
Epoch [22/100], Loss: 0.016141305981203915, Train Accuracy: 91.19 %,
Test Accuracy: 69.84 %
Epoch [23/100], Loss: 0.015479591189324855, Train Accuracy: 91.36 %,
Test Accuracy: 68.8 %
Epoch [24/100], Loss: 0.014489568036049604, Train Accuracy: 92.97 %,
Test Accuracy: 69.99 %
Epoch [25/100], Loss: 0.0143239476531744, Train Accuracy: 92.51 %,
Test Accuracy: 68.78 %
Epoch [26/100], Loss: 0.013528452199697494, Train Accuracy: 93.99 %,
Test Accuracy: 68.91 %
Epoch [27/100], Loss: 0.012187933020293713, Train Accuracy: 94.94 %,
Test Accuracy: 69.74 %
Epoch [28/100], Loss: 0.012099649068713188, Train Accuracy: 95.1 %,
Test Accuracy: 70.68 %
Epoch [29/100], Loss: 0.01207623085603118, Train Accuracy: 96.03 %,
Test Accuracy: 69.57 %
Epoch [30/100], Loss: 0.010783808449283241, Train Accuracy: 96.06 %,
Test Accuracy: 69.72 %
Epoch [31/100], Loss: 0.010600046747550369, Train Accuracy: 96.73 %,
Test Accuracy: 69.6 %
Epoch [32/100], Loss: 0.01025709756538272, Train Accuracy: 97.51 %,
Test Accuracy: 69.81 %
Epoch [33/100], Loss: 0.009383783822879195, Train Accuracy: 96.81 %,
Test Accuracy: 69.67 %
Epoch [34/100], Loss: 0.009868449518457055, Train Accuracy: 97.8 %,
Test Accuracy: 70.03 %
Epoch [35/100], Loss: 0.009189799649640918, Train Accuracy: 97.27 %,
Test Accuracy: 69.68 %
Epoch [36/100], Loss: 0.009272440471872687, Train Accuracy: 97.25 %,
Test Accuracy: 68.78 %
Epoch [37/100], Loss: 0.008511614551395178, Train Accuracy: 97.96 %,
Test Accuracy: 69.03 %
Epoch [38/100], Loss: 0.008448201079107822, Train Accuracy: 97.62 %,
Test Accuracy: 69.89 %
Epoch [39/100], Loss: 0.007921658363938332, Train Accuracy: 98.03 %,
Test Accuracy: 69.62 %
Epoch [40/100], Loss: 0.007469179351627827, Train Accuracy: 98.57 %,
Test Accuracy: 69.73 %
Epoch [41/100], Loss: 0.00804105860348791, Train Accuracy: 98.27 %,
Test Accuracy: 69.6 %
Epoch [42/100], Loss: 0.008028845932893455, Train Accuracy: 98.35 %,
Test Accuracy: 70.07 %
Epoch [43/100], Loss: 0.007354365881159902, Train Accuracy: 97.77 %,
Test Accuracy: 68.82 %
Epoch [44/100], Loss: 0.0070558781406842176, Train Accuracy: 98.81 %,
Test Accuracy: 69.52 %
```

```
Epoch [45/100], Loss: 0.006886252293922007, Train Accuracy: 98.6 %,
Test Accuracy: 69.05 %
Epoch [46/100], Loss: 0.006767110380716622, Train Accuracy: 98.45 %,
Test Accuracy: 69.34 %
Epoch [47/100], Loss: 0.006863414843752981, Train Accuracy: 98.79 %,
Test Accuracy: 68.18 %
Epoch [48/100], Loss: 0.006614585657324642, Train Accuracy: 98.3 %,
Test Accuracy: 68.14 %
Epoch [49/100], Loss: 0.006318155534751713, Train Accuracy: 99.16 %,
Test Accuracy: 69.74 %
Epoch [50/100], Loss: 0.006211050717066974, Train Accuracy: 99.02 %,
Test Accuracy: 68.68 %
Epoch [51/100], Loss: 0.006354795063938945, Train Accuracy: 98.94 %,
Test Accuracy: 69.11 %
Epoch [52/100], Loss: 0.00616648160610348, Train Accuracy: 99.01 %,
Test Accuracy: 69.43 %
Epoch [53/100], Loss: 0.005781056106230244, Train Accuracy: 98.97 %,
Test Accuracy: 68.96 %
Epoch [54/100], Loss: 0.0063973751722835, Train Accuracy: 98.64 %,
Test Accuracy: 69.11 %
Epoch [55/100], Loss: 0.00556597429709509, Train Accuracy: 98.99 %,
Test Accuracy: 69.3 %
Epoch [56/100], Loss: 0.004557068358827382, Train Accuracy: 98.75 %,
Test Accuracy: 69.11 %
Epoch [57/100], Loss: 0.005166569266747683, Train Accuracy: 98.86 %,
Test Accuracy: 69.4 %
Epoch [58/100], Loss: 0.005466851857490838, Train Accuracy: 99.26 %,
Test Accuracy: 69.94 %
Epoch [59/100], Loss: 0.005262756021483801, Train Accuracy: 98.84 %,
Test Accuracy: 68.76 %
Epoch [60/100], Loss: 0.004887941885087639, Train Accuracy: 99.2 %,
Test Accuracy: 68.61 %
Epoch [61/100], Loss: 0.004803817859105766, Train Accuracy: 99.27 %,
Test Accuracy: 69.5 %
Epoch [62/100], Loss: 0.0051346287813037635, Train Accuracy: 99.28 %,
Test Accuracy: 70.43 %
Epoch [63/100], Loss: 0.005268941349815578, Train Accuracy: 99.19 %,
Test Accuracy: 69.09 %
Epoch [64/100], Loss: 0.004708331418875605, Train Accuracy: 99.41 %,
Test Accuracy: 69.87 %
Epoch [65/100], Loss: 0.004666602420061827, Train Accuracy: 98.76 %,
Test Accuracy: 68.66 %
Epoch [66/100], Loss: 0.00484447438751813, Train Accuracy: 99.11 %,
Test Accuracy: 69.39 %
Epoch [67/100], Loss: 0.004747715530265123, Train Accuracy: 99.24 %,
Test Accuracy: 69.16 %
Epoch [68/100], Loss: 0.004648704421008005, Train Accuracy: 99.27 %,
Test Accuracy: 68.88 %
Epoch [69/100], Loss: 0.004660219957749359, Train Accuracy: 99.14 %,
Test Accuracy: 69.15 %
```

```
Epoch [70/100], Loss: 0.004390305569022894, Train Accuracy: 99.4 %,
Test Accuracy: 69.79 %
Epoch [71/100], Loss: 0.004135694066016003, Train Accuracy: 99.5 %,
Test Accuracy: 69.37 %
Epoch [72/100], Loss: 0.004240942480368539, Train Accuracy: 99.43 %,
Test Accuracy: 68.4 %
Epoch [73/100], Loss: 0.004264047497138381, Train Accuracy: 99.34 %,
Test Accuracy: 69.58 %
Epoch [74/100], Loss: 0.004127073148358613, Train Accuracy: 99.34 %,
Test Accuracy: 68.88 %
Epoch [75/100], Loss: 0.0040501711803488435, Train Accuracy: 99.43 %,
Test Accuracy: 69.26 %
Epoch [76/100], Loss: 0.003984583515347913, Train Accuracy: 99.49 %,
Test Accuracy: 68.9 %
Epoch [77/100], Loss: 0.004279429937060922, Train Accuracy: 99.3 %,
Test Accuracy: 69.05 %
Epoch [78/100], Loss: 0.00392565614641644, Train Accuracy: 99.35 %,
Test Accuracy: 68.96 %
Epoch [79/100], Loss: 0.004141333803907037, Train Accuracy: 99.57 %,
Test Accuracy: 69.13 %
Epoch [80/100], Loss: 0.004078554537310265, Train Accuracy: 99.41 %,
Test Accuracy: 68.52 %
Epoch [81/100], Loss: 0.004162466019357089, Train Accuracy: 99.46 %,
Test Accuracy: 69.72 %
Epoch [82/100], Loss: 0.0034878421830479055, Train Accuracy: 99.3 %,
Test Accuracy: 68.82 %
Epoch [83/100], Loss: 0.003969288804382086, Train Accuracy: 99.39 %,
Test Accuracy: 68.53 %
Epoch [84/100], Loss: 0.0037406712533906102, Train Accuracy: 99.32 %,
Test Accuracy: 69.44 %
Epoch [85/100], Loss: 0.004018598759174347, Train Accuracy: 99.49 %,
Test Accuracy: 68.33 %
Epoch [86/100], Loss: 0.0029845897213439457, Train Accuracy: 99.5 %,
Test Accuracy: 69.5 %
Epoch [87/100], Loss: 0.002871922108042054, Train Accuracy: 99.57 %,
Test Accuracy: 68.79 %
Epoch [88/100], Loss: 0.003449461415456608, Train Accuracy: 99.52 %,
Test Accuracy: 69.61 %
Epoch [89/100], Loss: 0.0030828032821184023, Train Accuracy: 99.48 %,
Test Accuracy: 69.53 %
Epoch [90/100], Loss: 0.0035288139774813316, Train Accuracy: 99.46 %,
Test Accuracy: 69.62 %
Epoch [91/100], Loss: 0.0032943236007587982, Train Accuracy: 99.33 %,
Test Accuracy: 68.07 %
Epoch [92/100], Loss: 0.003340856035600882, Train Accuracy: 99.57 %,
Test Accuracy: 69.61 %
Epoch [93/100], Loss: 0.0031475952598499133, Train Accuracy: 99.65 %,
Test Accuracy: 69.34 %
Epoch [94/100], Loss: 0.0033332297490735073, Train Accuracy: 99.59 %,
Test Accuracy: 68.72 %
```

```
Epoch [95/100], Loss: 0.003452564359817188, Train Accuracy: 99.64 %,
Test Accuracy: 69.64 %
Epoch [96/100], Loss: 0.0033024775293190034, Train Accuracy: 99.61 %,
Test Accuracy: 68.93 %
Epoch [97/100], Loss: 0.00331978090062039, Train Accuracy: 99.61 %,
Test Accuracy: 68.91 %
Epoch [98/100], Loss: 0.002841195673972834, Train Accuracy: 99.73 %,
Test Accuracy: 69.48 %
Epoch [99/100], Loss: 0.003509043972881045, Train Accuracy: 99.65 %,
Test Accuracy: 69.36 %
Epoch [100/100], Loss: 0.003461717249144567, Train Accuracy: 99.39 %,
Test Accuracy: 68.69 %
```



```python
print('Test Accuracy for Cutout:', test_accuracy)
```

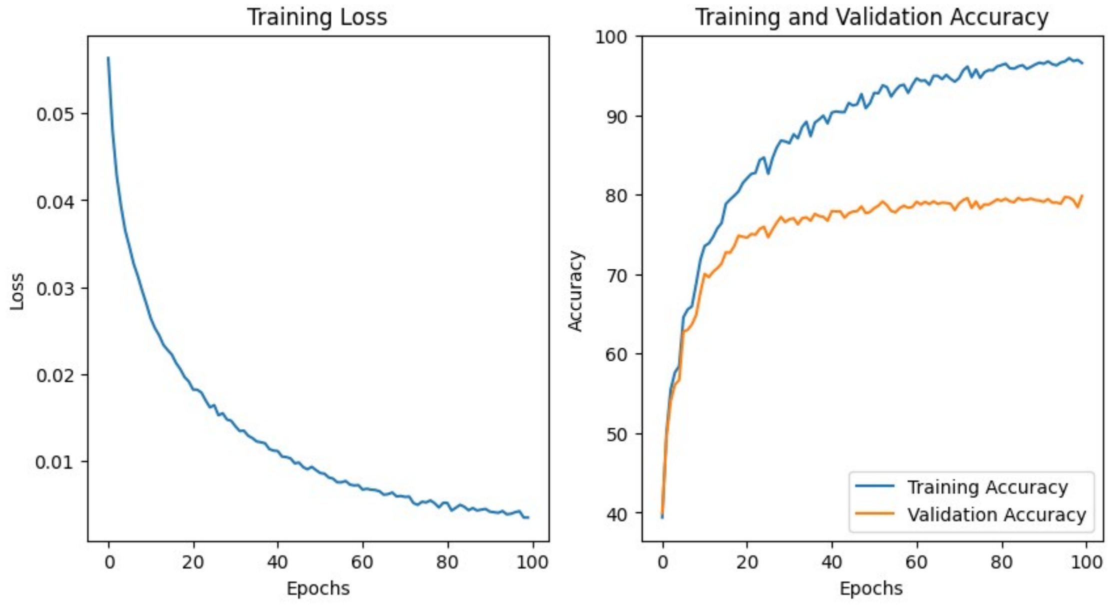
```
Test Accuracy for Cutout: 68.69
```

Standard Augmentation

```python
train_loader = DataLoader(train_data, batch_size=32, sampler=sampler)
test_loader = DataLoader(test_data, batch_size=32)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
# Create an instance of the ResNet20 model
model = ResNet20()
model = model.to(device)  # move the model to GPU if available

# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

```
class StandardAugmentation(object):
    def __init__(self, size=32, padding=4, p_flip=0.5):
        self.size = size
        self.padding = padding
        self.p_flip = p_flip

    def __call__(self, img):
        # Add padding
        padded_img = torch.zeros((img.shape[0], img.shape[1],
img.shape[2] + 2 * self.padding, img.shape[3] + 2 * self.padding))
        padded_img[:, :, self.padding: -self.padding, self.padding: -
self.padding] = img

        # Random crop
        x = random.randint(0, padded_img.shape[-2] - self.size)
        y = random.randint(0, padded_img.shape[-1] - self.size)
        cropped_img = padded_img[:, :, x:x+self.size, y:y+self.size]

        # Random horizontal flip
        flipped_img = cropped_img
        if random.random() < self.p_flip:
            for i in range(img.shape[0]):
                flipped_img[i] = flipped_img[i].flip(-1)

        return flipped_img

standard_augmentation = StandardAugmentation(size=32, padding=4,
p_flip=0.5)


train_accuracies = []
test_accuracies = []
losses = []
num_epochs=100
for epoch in range(num_epochs):
    model.train()
    epoch_loss = 0
    for i, data in enumerate(train_loader):
        images, labels = data
        images = standard_augmentation(images)
        images = images.to(device)
        labels = labels.to(device)

        # zero the parameter gradients
        optimizer.zero_grad()

        # forward pass
        outputs = model(images)
        loss = criterion(outputs, labels)
        epoch_loss += loss.item()
        # backward pass and optimization
        loss.backward()
        optimizer.step()
    train_accuracy,total = accuracy(model, train_loader)
    test_accuracy ,total= accuracy(model, test_loader)
    train_accuracies.append(train_accuracy)
    test_accuracies.append(test_accuracy)

    losses.append(epoch_loss / total)


    print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {epoch_loss /
total}, Train Accuracy: {train_accuracy} %, Test Accuracy:
{test_accuracy} %')

plt.figure(figsize=(10,5))
plt.subplot(1, 2, 1)
plt.plot(losses)
plt.title('Training Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')



plt.subplot(1, 2, 2)
plt.plot(train_accuracies, label='Training Accuracy')
plt.plot(test_accuracies, label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

```
Epoch [1/100], Loss: 0.05634603952169418, Train Accuracy: 39.37 %,
Test Accuracy: 39.92 %
Epoch [2/100], Loss: 0.048086394584178925, Train Accuracy: 49.97 %,
Test Accuracy: 49.51 %
Epoch [3/100], Loss: 0.0428044737637043, Train Accuracy: 55.49 %, Test
Accuracy: 54.03 %
Epoch [4/100], Loss: 0.039332235425710675, Train Accuracy: 57.64 %,
Test Accuracy: 56.1 %
Epoch [5/100], Loss: 0.036509343165159226, Train Accuracy: 58.44 %,
Test Accuracy: 56.66 %
Epoch [6/100], Loss: 0.03464281345009804, Train Accuracy: 64.58 %,
Test Accuracy: 62.78 %
Epoch [7/100], Loss: 0.032657340556383134, Train Accuracy: 65.56 %,
Test Accuracy: 62.99 %
Epoch [8/100], Loss: 0.031221652084589005, Train Accuracy: 65.95 %,
Test Accuracy: 63.68 %
Epoch [9/100], Loss: 0.029571286565065386, Train Accuracy: 68.78 %,
Test Accuracy: 64.85 %
Epoch [10/100], Loss: 0.028050469279289246, Train Accuracy: 71.8 %,
Test Accuracy: 67.66 %
Epoch [11/100], Loss: 0.026416957798600198, Train Accuracy: 73.55 %,
Test Accuracy: 70.04 %
Epoch [12/100], Loss: 0.02527592375576496, Train Accuracy: 73.88 %,
Test Accuracy: 69.62 %
Epoch [13/100], Loss: 0.024452181577682496, Train Accuracy: 74.69 %,
Test Accuracy: 70.3 %
Epoch [14/100], Loss: 0.02334833567738533, Train Accuracy: 75.77 %,
Test Accuracy: 70.75 %
Epoch [15/100], Loss: 0.022750776921212672, Train Accuracy: 76.43 %,
Test Accuracy: 71.33 %
Epoch [16/100], Loss: 0.022214392161369324, Train Accuracy: 78.84 %,
Test Accuracy: 72.75 %
Epoch [17/100], Loss: 0.02124829646348953, Train Accuracy: 79.39 %,
Test Accuracy: 72.69 %
Epoch [18/100], Loss: 0.020546356026828288, Train Accuracy: 79.88 %,
Test Accuracy: 73.55 %
Epoch [19/100], Loss: 0.01963191700875759, Train Accuracy: 80.43 %,
Test Accuracy: 74.86 %
Epoch [20/100], Loss: 0.019126475781202315, Train Accuracy: 81.46 %,
Test Accuracy: 74.71 %
Epoch [21/100], Loss: 0.018196523788571357, Train Accuracy: 82.06 %,
Test Accuracy: 74.59 %
Epoch [22/100], Loss: 0.01817522464245558, Train Accuracy: 82.61 %,
Test Accuracy: 75.05 %
Epoch [23/100], Loss: 0.01783224593400955, Train Accuracy: 82.76 %,
Test Accuracy: 74.97 %
Epoch [24/100], Loss: 0.016924155934154986, Train Accuracy: 84.36 %,
Test Accuracy: 75.7 %
Epoch [25/100], Loss: 0.016139284940063954, Train Accuracy: 84.68 %,
Test Accuracy: 75.96 %
Epoch [26/100], Loss: 0.01640768565386534, Train Accuracy: 82.63 %,
Test Accuracy: 74.64 %
Epoch [27/100], Loss: 0.015228735329210758, Train Accuracy: 84.58 %,
Test Accuracy: 75.62 %
Epoch [28/100], Loss: 0.015469031517207623, Train Accuracy: 85.93 %,
Test Accuracy: 76.48 %
Epoch [29/100], Loss: 0.01474360561221838, Train Accuracy: 86.81 %,
Test Accuracy: 77.21 %
Epoch [30/100], Loss: 0.014600658370554447, Train Accuracy: 86.71 %,
Test Accuracy: 76.55 %
Epoch [31/100], Loss: 0.013966659054160118, Train Accuracy: 86.48 %,
Test Accuracy: 76.91 %
Epoch [32/100], Loss: 0.013420660872757435, Train Accuracy: 87.6 %,
Test Accuracy: 77.03 %
Epoch [33/100], Loss: 0.013456285867094993, Train Accuracy: 87.09 %,
Test Accuracy: 76.27 %
Epoch [34/100], Loss: 0.012882958045601845, Train Accuracy: 88.51 %,
Test Accuracy: 77.04 %
Epoch [35/100], Loss: 0.012608466505259276, Train Accuracy: 89.19 %,
Test Accuracy: 77.15 %
Epoch [36/100], Loss: 0.012221072363108397, Train Accuracy: 87.36 %,
Test Accuracy: 76.73 %
Epoch [37/100], Loss: 0.012128077284991741, Train Accuracy: 89.07 %,
Test Accuracy: 77.59 %
Epoch [38/100], Loss: 0.01200998683422804, Train Accuracy: 89.47 %,
Test Accuracy: 77.31 %
Epoch [39/100], Loss: 0.011335987974703312, Train Accuracy: 89.95 %,
Test Accuracy: 77.21 %
Epoch [40/100], Loss: 0.0111716813005507, Train Accuracy: 88.97 %,
Test Accuracy: 76.73 %
Epoch [41/100], Loss: 0.011109628909826278, Train Accuracy: 90.34 %,
Test Accuracy: 77.94 %
Epoch [42/100], Loss: 0.01047668211609125, Train Accuracy: 90.48 %,
Test Accuracy: 77.89 %
Epoch [43/100], Loss: 0.010422131188586354, Train Accuracy: 90.4 %,
Test Accuracy: 77.91 %
Epoch [44/100], Loss: 0.010267122273519636, Train Accuracy: 90.39 %,
Test Accuracy: 77.1 %
Epoch [45/100], Loss: 0.009705564376339317, Train Accuracy: 91.55 %,
Test Accuracy: 77.65 %
Epoch [46/100], Loss: 0.009789479262009263, Train Accuracy: 91.22 %,
Test Accuracy: 77.88 %
Epoch [47/100], Loss: 0.0092584667570889, Train Accuracy: 91.38 %,
Test Accuracy: 77.93 %
Epoch [48/100], Loss: 0.00901382887735963, Train Accuracy: 92.68 %,
Test Accuracy: 78.52 %
Epoch [49/100], Loss: 0.009299277547746896, Train Accuracy: 90.88 %,
Test Accuracy: 77.68 %
Epoch [50/100], Loss: 0.00892719032652676, Train Accuracy: 91.52 %,
Test Accuracy: 77.82 %
Epoch [51/100], Loss: 0.008586114860326052, Train Accuracy: 92.82 %,
Test Accuracy: 78.3 %
Epoch [52/100], Loss: 0.008513476834073663, Train Accuracy: 92.77 %,
Test Accuracy: 78.62 %
Epoch [53/100], Loss: 0.008071583807840944, Train Accuracy: 93.77 %,
Test Accuracy: 79.15 %
Epoch [54/100], Loss: 0.007953053579106927, Train Accuracy: 93.54 %,
Test Accuracy: 78.67 %
Epoch [55/100], Loss: 0.007519290852546692, Train Accuracy: 92.34 %,
Test Accuracy: 77.96 %
Epoch [56/100], Loss: 0.0075169523019343615, Train Accuracy: 93.12 %,
Test Accuracy: 77.77 %
Epoch [57/100], Loss: 0.007667454134300351, Train Accuracy: 93.7 %,
Test Accuracy: 78.3 %
Epoch [58/100], Loss: 0.007257724864035845, Train Accuracy: 93.85 %,
Test Accuracy: 78.62 %
Epoch [59/100], Loss: 0.007152441827394068, Train Accuracy: 92.81 %,
Test Accuracy: 78.35 %
Epoch [60/100], Loss: 0.007195025129988789 6, Train Accuracy: 93.82 %,
Test Accuracy: 78.45 %
Epoch [61/100], Loss: 0.006681256479769945, Train Accuracy: 94.65 %,
Test Accuracy: 79.12 %
Epoch [62/100], Loss: 0.006776556279696524, Train Accuracy: 94.32 %,
Test Accuracy: 78.76 %
Epoch [63/100], Loss: 0.006667059365659952, Train Accuracy: 94.39 %,
Test Accuracy: 79.1 %
Epoch [64/100], Loss: 0.006625545059330762, Train Accuracy: 93.83 %,
Test Accuracy: 78.82 %
Epoch [65/100], Loss: 0.006483865377493203, Train Accuracy: 94.97 %,
Test Accuracy: 79.16 %
Epoch [66/100], Loss: 0.006087245870754123, Train Accuracy: 94.96 %,
Test Accuracy: 78.84 %
Epoch [67/100], Loss: 0.006162066694162786, Train Accuracy: 94.54 %,
Test Accuracy: 79.0 %
Epoch [68/100], Loss: 0.006339578060992062, Train Accuracy: 95.08 %,
Test Accuracy: 78.95 %
Epoch [69/100], Loss: 0.005892068083025515, Train Accuracy: 94.59 %,
Test Accuracy: 78.88 %
Epoch [70/100], Loss: 0.005946931722387671, Train Accuracy: 94.22 %,
Test Accuracy: 78.07 %
Epoch [71/100], Loss: 0.005835959081351757, Train Accuracy: 94.66 %,
Test Accuracy: 78.88 %
Epoch [72/100], Loss: 0.005869266271591187, Train Accuracy: 95.65 %,
Test Accuracy: 79.32 %
Epoch [73/100], Loss: 0.0051263767299242315, Train Accuracy: 96.13 %,
Test Accuracy: 79.57 %
Epoch [74/100], Loss: 0.004922295379266143, Train Accuracy: 94.78 %,
Test Accuracy: 78.33 %
Epoch [75/100], Loss: 0.005278869095910341, Train Accuracy: 95.76 %,
Test Accuracy: 79.15 %
Epoch [76/100], Loss: 0.005199574086349457, Train Accuracy: 94.69 %,
Test Accuracy: 78.24 %
Epoch [77/100], Loss: 0.005437367359735072, Train Accuracy: 95.41 %,
Test Accuracy: 78.79 %
Epoch [78/100], Loss: 0.005105032235942781, Train Accuracy: 95.68 %,
Test Accuracy: 78.75 %
Epoch [79/100], Loss: 0.004615730417892337, Train Accuracy: 95.66 %,
Test Accuracy: 79.06 %
Epoch [80/100], Loss: 0.0051500379649922256, Train Accuracy: 96.14 %,
Test Accuracy: 79.4 %
Epoch [81/100], Loss: 0.005131516703777015, Train Accuracy: 96.3 %,
Test Accuracy: 79.23 %
Epoch [82/100], Loss: 0.004262567760143429, Train Accuracy: 96.49 %,
Test Accuracy: 79.5 %
Epoch [83/100], Loss: 0.004586641840264201, Train Accuracy: 95.91 %,
Test Accuracy: 79.17 %
Epoch [84/100], Loss: 0.004925183708406985, Train Accuracy: 95.86 %,
Test Accuracy: 79.05 %
Epoch [85/100], Loss: 0.004695371745154261‌5, Train Accuracy: 96.15 %,
Test Accuracy: 79.6 %
Epoch [86/100], Loss: 0.004302178014628589, Train Accuracy: 96.3 %,
Test Accuracy: 79.31 %
Epoch [87/100], Loss: 0.004552656112238765, Train Accuracy: 95.83 %,
Test Accuracy: 79.37 %
Epoch [88/100], Loss: 0.004268033773917705, Train Accuracy: 96.09 %,
Test Accuracy: 79.51 %
Epoch [89/100], Loss: 0.004372336038760841, Train Accuracy: 96.39 %,
Test Accuracy: 79.31 %
Epoch [90/100], Loss: 0.00445322365090251, Train Accuracy: 96.62 %,
Test Accuracy: 79.26 %
Epoch [91/100], Loss: 0.0041429180128499865, Train Accuracy: 96.48 %,
Test Accuracy: 79.1 %
Epoch [92/100], Loss: 0.0040742193093523385, Train Accuracy: 96.78 %,
Test Accuracy: 79.45 %
Epoch [93/100], Loss: 0.00400455097197555, Train Accuracy: 96.43 %,
Test Accuracy: 79.0 %
Epoch [94/100], Loss: 0.004209936984907836, Train Accuracy: 96.27 %,
Test Accuracy: 79.02 %
Epoch [95/100], Loss: 0.0038330578430555763, Train Accuracy: 96.63 %,
Test Accuracy: 78.86 %
Epoch [96/100], Loss: 0.0038963230235967783, Train Accuracy: 96.78 %,
Test Accuracy: 79.71 %
Epoch [97/100], Loss: 0.004078137434273958, Train Accuracy: 97.19 %,
Test Accuracy: 79.65 %
Epoch [98/100], Loss: 0.004200498043373227, Train Accuracy: 96.81 %,
Test Accuracy: 79.31 %
Epoch [99/100], Loss: 0.003463858882896602, Train Accuracy: 96.97 %,
Test Accuracy: 78.42 %
Epoch [100/100], Loss: 0.0034571039682254195, Train Accuracy: 96.57 %,
Test Accuracy: 79.84 %
```



```
print('Test Accuracy for Standard Aug:', test_accuracy)
```

```
Test Accuracy for Standard Aug: 79.84
```

Combining all Augmentations

```
train_loader = DataLoader(train_data, batch_size=32, sampler=sampler)
test_loader = DataLoader(test_data, batch_size=32)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
# Create an instance of the ResNet20 model
model = ResNet20()
model = model.to(device)

# Define the loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)
```

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

# Define the augmentations
cutout_aug = Cutout(n_holes=1, length=16)
standard_aug = StandardAugmentation(size=32, padding=4, p_flip=0.5)

n_epochs = 100
train_accuracies=[]
test_accuracies=[]
losses=[]
for epoch in range(n_epochs):
    train_loss = 0.0
    correct = 0
    total = 0
    model.train()
    for i, (inputs, labels) in (enumerate(train_loader)):
        # Apply the augmentations
        augmented_inputs = [cutout_aug(standard_aug(img)) for img in
inputs]
        inputs = torch.stack(augmented_inputs).to(device)
        labels = labels.to(device)


        inputs, labels_a, labels_b, lam = mixup_data(inputs, labels,
alpha=0.4)
        labels_a = labels_a.long().to(device)
        labels_b = labels_b.long().to(device)
        optimizer.zero_grad()

        # Forward pass
        outputs = model(inputs)
        loss_a = criterion(outputs, labels_a)
        loss_b = criterion(outputs, labels_b)
        loss = lam * loss_a + (1 - lam) * loss_b
        train_loss += loss.item()

        # Backward and optimize
        loss.backward()
        optimizer.step()
    train_accuracy ,total= accuracy(model,train_loader)
    train_accuracies.append(train_accuracy)
    test_accuracy ,total= accuracy(model,test_loader)
    test_accuracies.append(test_accuracy)
    losses.append(train_loss/total)
    print(f"Epoch {epoch+1}/{n_epochs}, Loss: {train_loss/total},
Train Accuracy: {train_accuracy}, Test Accuracy: {test_accuracy}")
```

```
Epoch 1/100, Loss: 0.07069858858585358, Train Accuracy: 29.86, Test
Accuracy: 29.7
Epoch 2/100, Loss: 0.06989343636035919, Train Accuracy: 31.33, Test
Accuracy: 31.72
Epoch 3/100, Loss: 0.06916219555139541, Train Accuracy: 33.52, Test
Accuracy: 33.33
Epoch 4/100, Loss: 0.06908356088399888, Train Accuracy: 37.95, Test
Accuracy: 37.24
Epoch 5/100, Loss: 0.06809376275539399, Train Accuracy: 38.86, Test
Accuracy: 38.33
Epoch 6/100, Loss: 0.06807328803539277, Train Accuracy: 42.52, Test
Accuracy: 42.18
Epoch 7/100, Loss: 0.06741181067228318, Train Accuracy: 43.01, Test
Accuracy: 42.23
Epoch 8/100, Loss: 0.06738780657052994, Train Accuracy: 43.53, Test
Accuracy: 43.16
Epoch 9/100, Loss: 0.06777063442468644, Train Accuracy: 43.3, Test
Accuracy: 42.57
```

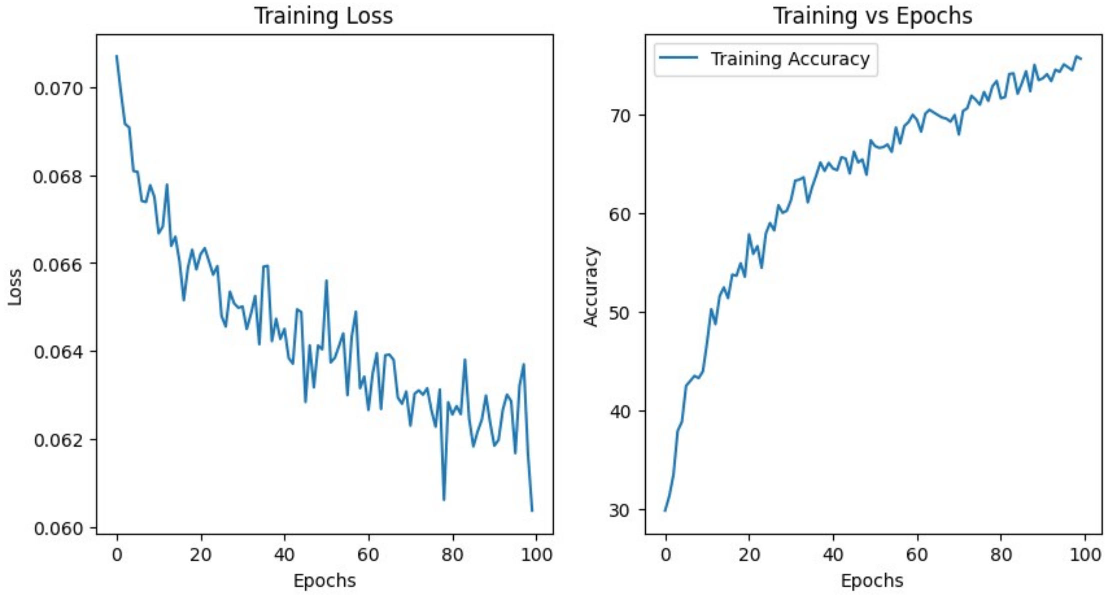
```
Epoch 10/100, Loss: 0.06750144842863083, Train Accuracy: 43.98, Test
Accuracy: 43.24
Epoch 11/100, Loss: 0.06667621711492538, Train Accuracy: 46.92, Test
Accuracy: 46.17
Epoch 12/100, Loss: 0.06683182903528213, Train Accuracy: 50.3, Test
Accuracy: 49.38
Epoch 13/100, Loss: 0.06778328459262847, Train Accuracy: 48.77, Test
Accuracy: 48.7
Epoch 14/100, Loss: 0.06638967362642288, Train Accuracy: 51.62, Test
Accuracy: 51.15
Epoch 15/100, Loss: 0.06660102899074555, Train Accuracy: 52.5, Test
Accuracy: 50.66
Epoch 16/100, Loss: 0.06602400413751602, Train Accuracy: 51.41, Test
Accuracy: 50.91
Epoch 17/100, Loss: 0.06515540672540665, Train Accuracy: 53.8, Test
Accuracy: 52.79
Epoch 18/100, Loss: 0.06590195028781891, Train Accuracy: 53.68, Test
Accuracy: 51.97
Epoch 19/100, Loss: 0.066304087972641, Train Accuracy: 54.95, Test
Accuracy: 54.18
Epoch 20/100, Loss: 0.06585607945919036, Train Accuracy: 53.58, Test
Accuracy: 52.23
Epoch 21/100, Loss: 0.06620135787725448, Train Accuracy: 57.88, Test
Accuracy: 56.35
Epoch 22/100, Loss: 0.06633947474956513, Train Accuracy: 55.9, Test
Accuracy: 53.47
Epoch 23/100, Loss: 0.06604203284978867, Train Accuracy: 56.69, Test
Accuracy: 54.78
Epoch 24/100, Loss: 0.0657346894145012, Train Accuracy: 54.49, Test
Accuracy: 53.17
Epoch 25/100, Loss: 0.06593230175971986, Train Accuracy: 57.96, Test
Accuracy: 56.9
Epoch 26/100, Loss: 0.06479714325666427, Train Accuracy: 59.03, Test
Accuracy: 56.94
Epoch 27/100, Loss: 0.06455661273002625, Train Accuracy: 58.28, Test
Accuracy: 56.21
Epoch 28/100, Loss: 0.06535019830465316, Train Accuracy: 60.83, Test
Accuracy: 58.3
Epoch 29/100, Loss: 0.06508263731002807, Train Accuracy: 60.06, Test
Accuracy: 57.14
Epoch 30/100, Loss: 0.06498396767377854, Train Accuracy: 60.28, Test
Accuracy: 58.64
Epoch 31/100, Loss: 0.06501330357789993, Train Accuracy: 61.36, Test
Accuracy: 59.31
Epoch 32/100, Loss: 0.06450018693208695, Train Accuracy: 63.33, Test
Accuracy: 60.67
Epoch 33/100, Loss: 0.06484338842630387, Train Accuracy: 63.43, Test
Accuracy: 60.82
Epoch 34/100, Loss: 0.06525182610750198, Train Accuracy: 63.67, Test
Accuracy: 60.74
```

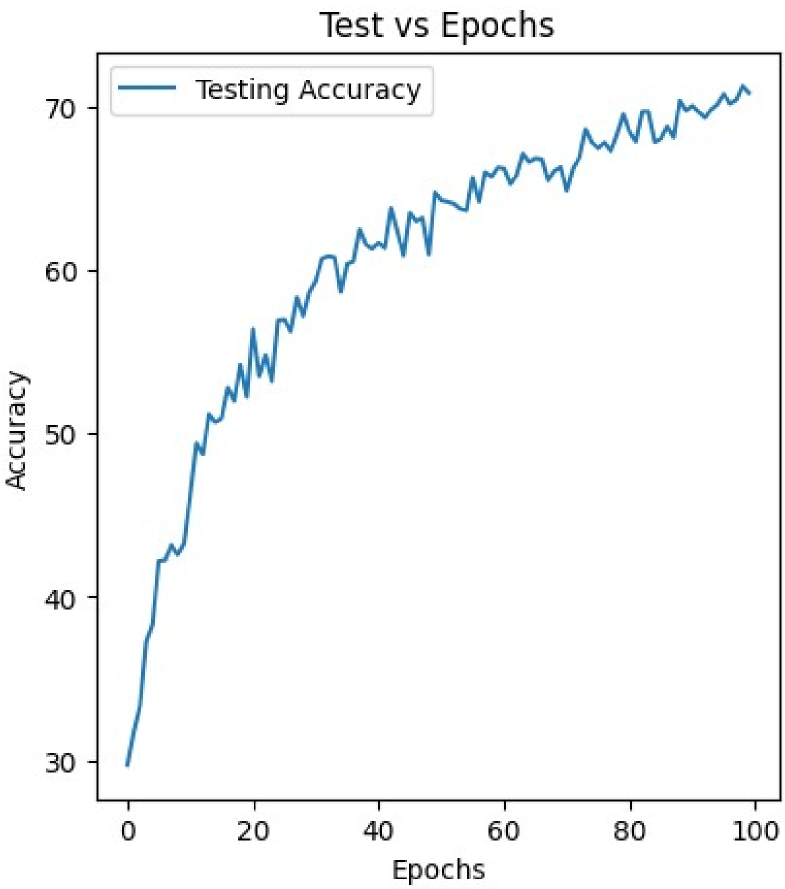
```
Epoch 35/100, Loss: 0.06415504968166351, Train Accuracy: 61.12, Test
Accuracy: 58.62
Epoch 36/100, Loss: 0.06591981855630874, Train Accuracy: 62.64, Test
Accuracy: 60.34
Epoch 37/100, Loss: 0.06593785059452056, Train Accuracy: 63.85, Test
Accuracy: 60.49
Epoch 38/100, Loss: 0.0642257261276245, Train Accuracy: 65.17, Test
Accuracy: 62.45
Epoch 39/100, Loss: 0.0647344030380249, Train Accuracy: 64.31, Test
Accuracy: 61.52
Epoch 40/100, Loss: 0.06427524063587188, Train Accuracy: 65.13, Test
Accuracy: 61.27
Epoch 41/100, Loss: 0.06450433348417282, Train Accuracy: 64.56, Test
Accuracy: 61.64
Epoch 42/100, Loss: 0.06384074034690856, Train Accuracy: 64.39, Test
Accuracy: 61.31
Epoch 43/100, Loss: 0.06371270365715027, Train Accuracy: 65.7, Test
Accuracy: 63.77
Epoch 44/100, Loss: 0.06494802947044373, Train Accuracy: 65.57, Test
Accuracy: 62.31
Epoch 45/100, Loss: 0.06488551052808762, Train Accuracy: 64.05, Test
Accuracy: 60.86
Epoch 46/100, Loss: 0.06284185909032822, Train Accuracy: 66.28, Test
Accuracy: 63.46
Epoch 47/100, Loss: 0.0641296576499939, Train Accuracy: 65.17, Test
Accuracy: 62.95
Epoch 48/100, Loss: 0.0631743458032608, Train Accuracy: 65.48, Test
Accuracy: 63.16
Epoch 49/100, Loss: 0.06412671191692353, Train Accuracy: 63.92, Test
Accuracy: 60.93
Epoch 50/100, Loss: 0.06403904106616974, Train Accuracy: 67.42, Test
Accuracy: 64.72
Epoch 51/100, Loss: 0.06560108251571656, Train Accuracy: 66.84, Test
Accuracy: 64.23
Epoch 52/100, Loss: 0.06374086184501648, Train Accuracy: 66.65, Test
Accuracy: 64.14
Epoch 53/100, Loss: 0.0638475995182991, Train Accuracy: 66.73, Test
Accuracy: 64.03
Epoch 54/100, Loss: 0.06411898157596588, Train Accuracy: 67.0, Test
Accuracy: 63.72
Epoch 55/100, Loss: 0.06440221983194351, Train Accuracy: 66.24, Test
Accuracy: 63.6
Epoch 56/100, Loss: 0.06299686077833176, Train Accuracy: 68.72, Test
Accuracy: 65.62
Epoch 57/100, Loss: 0.06431732808351517, Train Accuracy: 67.09, Test
Accuracy: 64.15
Epoch 58/100, Loss: 0.06489630708694458, Train Accuracy: 68.87, Test
Accuracy: 65.95
Epoch 59/100, Loss: 0.06315499835014343, Train Accuracy: 69.27, Test
Accuracy: 65.67
```

```
Epoch 60/100, Loss: 0.0634215022444725, Train Accuracy: 70.01, Test
Accuracy: 66.23
Epoch 61/100, Loss: 0.06266143120527268, Train Accuracy: 69.5, Test
Accuracy: 66.18
Epoch 62/100, Loss: 0.06348886848688126, Train Accuracy: 68.29, Test
Accuracy: 65.24
Epoch 63/100, Loss: 0.06395071015357971, Train Accuracy: 70.12, Test
Accuracy: 65.76
Epoch 64/100, Loss: 0.062683624958992, Train Accuracy: 70.52, Test
Accuracy: 67.08
Epoch 65/100, Loss: 0.06390367295742035, Train Accuracy: 70.24, Test
Accuracy: 66.56
Epoch 66/100, Loss: 0.06391977890729904, Train Accuracy: 69.98, Test
Accuracy: 66.8
Epoch 67/100, Loss: 0.06379936003684998, Train Accuracy: 69.73, Test
Accuracy: 66.74
Epoch 68/100, Loss: 0.06294284467697144, Train Accuracy: 69.62, Test
Accuracy: 65.45
Epoch 69/100, Loss: 0.06280112283229827, Train Accuracy: 69.31, Test
Accuracy: 66.02
Epoch 70/100, Loss: 0.06308065061569214, Train Accuracy: 69.99, Test
Accuracy: 66.26
Epoch 71/100, Loss: 0.06230458112955094, Train Accuracy: 68.0, Test
Accuracy: 64.82
Epoch 72/100, Loss: 0.06303110319375992, Train Accuracy: 70.4, Test
Accuracy: 66.17
Epoch 73/100, Loss: 0.06310604399442672, Train Accuracy: 70.67, Test
Accuracy: 66.85
Epoch 74/100, Loss: 0.063006245803833, Train Accuracy: 71.94, Test
Accuracy: 68.58
Epoch 75/100, Loss: 0.06315518947839736, Train Accuracy: 71.54, Test
Accuracy: 67.76
Epoch 76/100, Loss: 0.06265629933476448, Train Accuracy: 71.03, Test
Accuracy: 67.41
Epoch 77/100, Loss: 0.06227827535867691, Train Accuracy: 72.32, Test
Accuracy: 67.77
Epoch 78/100, Loss: 0.06312569434642792, Train Accuracy: 71.41, Test
Accuracy: 67.24
Epoch 79/100, Loss: 0.060617355042696, Train Accuracy: 72.9, Test
Accuracy: 68.26
Epoch 80/100, Loss: 0.06283831799030304, Train Accuracy: 73.44, Test
Accuracy: 69.5
Epoch 81/100, Loss: 0.06255668874979019, Train Accuracy: 71.67, Test
Accuracy: 68.4
Epoch 82/100, Loss: 0.06274905523061752, Train Accuracy: 71.81, Test
Accuracy: 67.83
Epoch 83/100, Loss: 0.06256476546525955, Train Accuracy: 74.14, Test
Accuracy: 69.66
Epoch 84/100, Loss: 0.06380499622821807, Train Accuracy: 74.21, Test
Accuracy: 69.66
```

```
Epoch 85/100, Loss: 0.062459397149086, Train Accuracy: 72.14, Test
Accuracy: 67.8
Epoch 86/100, Loss: 0.06183049063682556, Train Accuracy: 73.21, Test
Accuracy: 68.0
Epoch 87/100, Loss: 0.06217051395177841, Train Accuracy: 74.43, Test
Accuracy: 68.75
Epoch 88/100, Loss: 0.062433197993040083, Train Accuracy: 72.39, Test
Accuracy: 68.08
Epoch 89/100, Loss: 0.06298867472410202, Train Accuracy: 75.08, Test
Accuracy: 70.32
Epoch 90/100, Loss: 0.06236801937222481, Train Accuracy: 73.52, Test
Accuracy: 69.71
Epoch 91/100, Loss: 0.06184811009168625, Train Accuracy: 73.71, Test
Accuracy: 70.01
Epoch 92/100, Loss: 0.06198018399477005, Train Accuracy: 74.13, Test
Accuracy: 69.64
Epoch 93/100, Loss: 0.0626580236196518, Train Accuracy: 73.42, Test
Accuracy: 69.29
Epoch 94/100, Loss: 0.06301061081886292, Train Accuracy: 74.58, Test
Accuracy: 69.77
Epoch 95/100, Loss: 0.0628612191081047, Train Accuracy: 74.36, Test
Accuracy: 70.09
Epoch 96/100, Loss: 0.06167726691961289, Train Accuracy: 75.12, Test
Accuracy: 70.74
Epoch 97/100, Loss: 0.06321063363552093, Train Accuracy: 74.83, Test
Accuracy: 70.13
Epoch 98/100, Loss: 0.06369994223713875, Train Accuracy: 74.53, Test
Accuracy: 70.37
Epoch 99/100, Loss: 0.06167135685682297, Train Accuracy: 75.93, Test
Accuracy: 71.22
Epoch 100/100, Loss: 0.06037219033837318, Train Accuracy: 75.69, Test
Accuracy: 70.81
```

```python
plt.figure(figsize=(10,5))
plt.subplot(1, 2, 1)
plt.plot(losses)
plt.title('Training Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')


plt.subplot(1, 2, 2)
plt.plot(train_accuracies, label='Training Accuracy')
plt.title('Training vs Epochs')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
plt.figure(figsize=(10,5))
plt.subplot(1, 2, 1)
plt.plot(test_accuracies, label='Testing Accuracy')
plt.title('Test vs Epochs')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```





```
print('Test Accuracy for Combined Aug:', test_accuracy)
```

```
Test Accuracy for Combined Aug: 70.81
```

Does combining improve things further? Combining different types of data augmentation techniques can potentially improve model performance further. It's vital to remember that it might also bring more computing complexity and cost. Additionally, over-augmentation could add extra noise or make the learning issue too complex, which might be detrimental to performance. As a result, it's essential to carefully select and fine-tune the augmentation methods in accordance with the particular task, dataset, and model architecture.

Q6) Data augmentation is essential for enhancing the effectiveness of machine learning models, particularly in the domain of image recognition tasks where the amount of available data may be limited.By performing a number of random transformations to the current data samples, such as rotation, scaling, translation, flipping, and other operations, data augmentation refers to the process of boosting the diversity and volume of training data. As a result, the model is exposed to a wider variety of data, which allows it to pick up more generalized features and patterns.In this assignment we worked on mixup,cutoff,and Standard augmentation techniques

Test accuracy: Data augmentation has the potential to increase test accuracy, particularly because it strengthens and improves the generalizability of the model. The model can handle different variations and anomalies in test data better by supplementing the training data with a variety of variations. In the evaluation step, this enables the model to generalize more effectively and obtain higher accuracy on untried samples. Train Accuracy: The impact of augmentation on train precision varies.Given that the enhanced data samples are different from the conventional or original data, we can first notice a decline in the train correctness.However, as the model is trained across a larger number of epochs, it is clear that the model learns from the augmented data. As a result, the train accuracy is significantly higher than in the scenario where no augmentation is used.Since augmented data is relatively fresh, it strengthens the model even further and enhances performance.

Is Test Accuracy Higher? In general, training accuracy will be higher than testing accuracy. The method by which we are changing the original data is the cause of our inability to achieve high accuracy. We transform the original data at every level before passing it on to the model, as opposed to merging the original data with augmented data. We are therefore getting comparable results both with and without augmentation.

Convergence of optimization: In terms of the convergence of optimisation, initialy since the augmented data is new and comlex, the model may take a longer amount of time to converge, but as the model keeps learning for more number of epochs, we see that it coverges faster, and can thus achieve better optimisation.The model will also avoid overfitting in the process.

Does training loss converge faster? Yes, Indeed the training loss decreased more rapidly for augmented data. The key reason behind this is that data augmentation techniques introduce the model to a broader and more varied collection of training instances. This increased diversity and volume of training data helps the model learn more robust features, allowing it to optimize and reduce loss more quickly.